

**ANÁLISIS Y DISEÑO DE UN CONTROLADOR DE ROBOTS  
EXPLORADORES TIPO ÁPODO**

**JONATHAN ALEXANDER BERMÚDEZ MORERA  
CLAUDIA TATIANA MARTÍNEZ SÁNCHEZ**

**UNIVERSIDAD PILOTO DE COLOMBIA  
FACULTAD INGENIERÍA MECATRÓNICA  
BOGOTÁ D.C**

**2016**

**ANÁLISIS Y DISEÑO DE UN CONTROLADOR DE ROBOTS  
EXPLORADORES TIPO ÁPODO**

**JONATHAN ALEXANDER BERMÚDEZ MORERA  
CLAUDIA TATIANA MARTÍNEZ SÁNCHEZ**

**MONOGRAFÍA DE TRABAJO DE GRADO PARA OPTAR AL  
TÍTULO DE INGENIERO MECATRÓNICO**

**Profesor orientador**

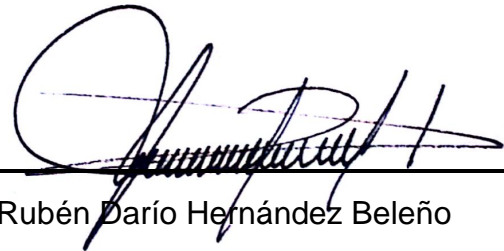
**MS.C ING. RUBÉN DARÍO HERNÁNDEZ BELEÑO PH.D (C)**

**UNIVERSIDAD PILOTO DE COLOMBIA  
FACULTAD INGENIERÍA MECATRÓNICA  
BOGOTÁ D.C**

**2016**

Nota de aceptación:

El trabajo de grado titulado  
**“ANÁLISIS Y DISEÑO DE UN  
CONTROLADOR DE ROBOTS  
EXPLORADORES TIPO ÁPODO”**  
elaborado por los estudiantes  
**Claudia Tatiana Martínez  
Sánchez y Jonathan Alexander  
Bermúdez Morera**, como requisito  
parcial para optar al título de  
Ingeniero Mecatrónico, fue  
aprobado por el Jurado Calificador



Rubén Darío Hernández Beleño  
Director del Proyecto



Claudia Tatiana Martínez Sánchez



Jonathan Alexander Bermúdez  
Morera

Bogotá D.C. 31 de Agosto de 2016

## DEDICATORIA

Dedico este trabajo a mis padres, mi hermana y en general a mi familia quienes a lo largo de mi vida me han acompañado tanto en mi formación académica como en mi formación personal, y quienes además me han brindado su apoyo incondicional, me han ayudado, aconsejado y acompañado durante este proceso. Gracias a ellos y a sus enseñanzas soy una persona integral y próximamente una futura profesional.

Tatiana Martínez S.

Dedico este trabajo a mi madre Esperanza Morera, mis hermanos y a mis amigos que con su apoyo y esfuerzo a lo largo de este camino que llamamos vida, me han ayudado a superar los obstáculos que se han presentado. Gracias a ellos ahora recorro mi camino con confianza, seguridad y mucha voluntad hacia un mejor futuro como una mejor persona y futuro profesional.

Jonathan Bermúdez M.



## AGRADECIMIENTOS

Primero quiero agradecer a Dios por brindarme una buena salud y capacidades para culminar este proceso de manera exitosa para ser una futura profesional y por darme la oportunidad de cumplir una meta más en mi vida. Además, quiero agradecer a mis padres, a mi hermana y en general a toda mi familia ya que ellos son mi motivación para salir adelante, son mi apoyo incondicional, son todo para mí. Agradezco a los ingenieros Rubén Darío Hernández y Oscar Avilés por ser nuestros guías, por colaborarnos, por brindarnos su apoyo, su experiencia y conocimientos ya que fueron de gran ayuda en el desarrollo de este proyecto. Finalmente, quiero agradecer a todo el grupo de Ingenieros que me enseñaron a lo largo de la carrera, gracias por la formación académica.

Tatiana Martínez S.

Yo agradezco a mi madre al ser mi apoyo económico y emocional en mi formación como persona y futuro profesional, también agradezco a mis hermanos que siempre han estado para brindarme su ayuda en los momentos que más he necesitado con sus consejos y sabiduría me han permitido desarrollarme como persona íntegra. Agradezco inmensamente a mis amigos que con su comprensión y consejos me han permitido superar obstáculos que en su momento consideré insuperables y de forma atenta doy mi agradecimiento y gratitud al ingeniero Rubén Darío Hernández y Oscar Avilés por guiarnos en este camino brindándonos sus conocimientos y su apoyo el cual nos permitió culminar con este trabajo; también agradezco al cuerpo docente que me ayudó a crecer tanto en mi formación personal, como futuro profesional.

Jonathan Bermúdez M.

# Índice general

<b>1. INTRODUCCIÓN</b>	<b>15</b>
1.1. Planteamiento y formulación del problema . . . . .	15
1.1.1. Descripción del problema . . . . .	16
1.1.2. Formulación del problema . . . . .	17
1.2. Justificación . . . . .	17
1.3. Estado del arte . . . . .	17
1.4. Objetivos . . . . .	23
1.4.1. General . . . . .	23
1.4.2. Específico . . . . .	24
1.5. Alcances y limitaciones . . . . .	24
1.6. Línea de investigación de programa . . . . .	24
1.7. Diseño metodológico . . . . .	24
1.8. Contribuciones . . . . .	25
<b>2. MARCO TEÓRICO</b>	<b>26</b>
2.1. Controladores . . . . .	29
2.1.1. PID (proporcional integrativo derivativo) . . . . .	29
2.1.1.1. Respuesta en frecuencia . . . . .	32
2.1.1.2. Reglas de Ziegler-Nichols para PID . . . . .	34
2.1.1.3. Lugar de las raíces . . . . .	38
2.1.2. Controlador PID en cascada . . . . .	41
2.1.3. Controlador PID discreto . . . . .	45
2.1.4. Controlador difuso . . . . .	49
2.1.5. Controlador PID-Fuzzy . . . . .	57
2.1.6. Comparación de controladores . . . . .	63
<b>3. MODELO MATEMÁTICO</b>	<b>65</b>
3.1. Locomoción robot ápedo . . . . .	65
3.2. Modelado del robot ápedo . . . . .	71
3.3. Cinemática directa . . . . .	75
3.3.1. Resultados cinemática directa . . . . .	77
3.4. Cinemática inversa . . . . .	80
3.4.1. Resultado de la cinemática inversa . . . . .	84

3.5. Velocidad angular . . . . .	86
<b>4. ESTUDIO DE MATERIAL</b>	<b>89</b>
4.1. Análisis dinámico de un módulo . . . . .	89
4.2. Volumen de pieza principal . . . . .	90
4.2.1. Implementación del volumen en la pieza . . . . .	94
4.3. Parámetros de material . . . . .	96
4.4. Materiales objeto de estudio . . . . .	97
4.4.1. Acero 1020 . . . . .	97
4.4.2. Aluminio . . . . .	98
4.4.3. ABS (acrilonitrilo butadieno estireno) . . . . .	98
4.4.4. Fibra de carbono . . . . .	99
4.5. Simulación de esfuerzos . . . . .	100
4.5.1. Acero 1020 . . . . .	100
4.5.2. Aluminio . . . . .	101
4.5.3. ABS . . . . .	102
4.6. Simulación de deformación . . . . .	102
4.6.1. Acero 1020 . . . . .	102
4.6.2. Aluminio . . . . .	103
4.6.3. ABS . . . . .	104
4.7. Resultados de materiales . . . . .	104
<b>5. RESULTADOS</b>	<b>106</b>
5.1. Modelo cinemático implementado . . . . .	106
5.2. Sistema lazo abierto . . . . .	107
5.2.1. Trayectoria . . . . .	107
5.2.2. Respuesta del modelo cinemático . . . . .	108
5.3. Diseño de controlador proporcional . . . . .	109
5.3.1. Resultados controlador proporcional . . . . .	111
5.4. Diseño de controlador difuso . . . . .	112
5.4.1. Variable de entrada y de salida . . . . .	113
5.4.2. Reglas difusas . . . . .	114
5.4.3. Resultados de control difuso . . . . .	116
5.4.3.1. Error resultante . . . . .	117
5.4.3.2. Trayectoria resultante . . . . .	117
5.5. Diseño de controlador proporcional-difuso . . . . .	118
5.5.1. Resultados del controlador proporcional-difuso . . . . .	121
5.6. Comparación de resultados . . . . .	124
<b>6. CONCLUSIONES Y TRABAJOS FUTUROS</b>	<b>126</b>
6.1. Conclusiones . . . . .	126
6.2. Trabajos futuros . . . . .	127

# Índice de figuras

1.1. Evolución de los robots serpiente de la familia ACM (Active Cord Mechanism) Hirose- Fukushima Robotics Lab . . . . .	16
1.2. Robots ápodos con movimiento en 2D . . . . .	18
1.3. Robot USAR para rescate . . . . .	18
1.4. Robots ápodos anfibios . . . . .	19
1.5. Robots ápodos escaladores . . . . .	20
1.6. Uncle Sam . . . . .	20
1.7. Robot ápodo LEGO . . . . .	21
1.8. Cubos en revolución . . . . .	21
1.9. Metodología utilizada . . . . .	25
2.1. Robots ápodos anfibios . . . . .	27
2.2. Robots ápodos trepadores . . . . .	27
2.3. Robot hiper-redundante trompa de elefante . . . . .	28
2.4. El robot ápodo Cube Reloaded . . . . .	29
2.5. Control PID . . . . .	30
2.6. Diagrama de PID . . . . .	31
2.7. Diagrama de <i>Bode</i> para $N(s) = \frac{4}{s(s^2+1)}$ . . . . .	33
2.8. Diagrama de <i>Bode</i> para $N(s) = \frac{4(5s+1)}{s(s^2+1)}$ . . . . .	33
2.9. Forma de la respuesta en forma de S . . . . .	34
2.10. Oscilación sostenida con periodo $P_{cr}$ en segundos . . . . .	36
2.11. Sistema lazo cerrado . . . . .	39
2.12. Polos y ceros de la planta . . . . .	39
2.13. Región de diseño . . . . .	40
2.14. PID lugar de las raíces . . . . .	40
2.15. Respuesta del sistema frente a un estímulo del tipo step . . . . .	41
2.16. Diagrama de bloques PID cascada . . . . .	42
2.17. PID cascada . . . . .	43
2.18. Respuesta frente PID en cascada . . . . .	44
2.19. Sistema lazo cerrado con PID digital . . . . .	46
2.20. Respuesta del sistema con el controlador PID discreto . . . . .	49
2.21. Controlador fuzzy genérico . . . . .	50
2.22. Arquitectura controlador fuzzy . . . . .	50

2.23. Reglas para el comportamiento de la velocidad de un motor . . . . .	51
2.24. Reglas para el comportamiento de la velocidad del motor . . . . .	51
2.25. Sistema de ejemplo . . . . .	52
2.26. Función de membresía $x$ . . . . .	52
2.27. Función de membresía en $y$ . . . . .	53
2.28. Reglas del sistema . . . . .	53
2.29. Regla #1 . . . . .	53
2.30. Regla #2 . . . . .	54
2.31. Regla #3 . . . . .	54
2.32. Regla # 1 evaluada en $x=2$ . . . . .	54
2.33. Regla # 2 evaluada en $x=2$ . . . . .	55
2.34. Regla # 3 evaluada en $x=2$ . . . . .	55
2.35. Conjunto resultante . . . . .	56
2.36. Conjunto resultante en « <i>Fuzzy</i> » . . . . .	56
2.37. Diagrama de bloques PID-fuzzy . . . . .	58
2.38. Controlador Fuzzy . . . . .	58
2.39. Funciones de pertenencia . . . . .	59
2.40. Arquitectura controlador híbrido PID-Fuzzy . . . . .	60
2.41. Primera simulación . . . . .	61
2.42. Salida de los controladores primera simulación . . . . .	61
2.43. Segunda simulación . . . . .	62
2.44. Salida de controladores segunda simulación . . . . .	63
3.1. El robot Miller y serpiente . . . . .	65
3.2. Gráfica de parámetros . . . . .	67
3.3. Onda de serpenteo con curvatura de $45^\circ$ en <i>Matlab</i> <sup>®</sup> . . . . .	69
3.4. Variación de ángulo de serpenteo . . . . .	70
3.5. Onda de serpenteo con curvatura de $121^\circ$ en <i>Matlab</i> <sup>®</sup> . . . . .	70
3.6. Desfase en la curva de serpenteo . . . . .	71
3.7. Gráfica en caso discreto . . . . .	71
3.8. Posición de los módulos . . . . .	73
3.9. Variación de ángulo de serpenteo en el plano discreto . . . . .	74
3.10. Onda de serpenteo con curvatura de 121 grados en el plano discreto en <i>Matlab</i> <sup>®</sup> . . . . .	74
3.11. Desfase en la curva de serpenteo . . . . .	75
3.12. Parámetros de la cinemática directa . . . . .	75
3.13. Modelo de robot ápedo . . . . .	76
3.14. Resultado de cinemática directa de un módulo . . . . .	78
3.15. Resultado de cinemática directa . . . . .	79
3.16. Parámetros de movimiento entre los módulos . . . . .	80
3.17. Posición entre módulos . . . . .	81
3.18. Resultado de cinemática inversa de un módulo . . . . .	84
3.19. Resultado de cinemática inversa. . . . .	85

3.20. Robot ápodo de cuatro eslabones . . . . .	86
4.1. Módulo de robot ápodo . . . . .	89
4.2. Pieza Principal . . . . .	91
4.3. Pieza principal -vista posterior . . . . .	91
4.4. Pieza principal-vista lateral. . . . .	92
4.5. Pieza principal acotada en milímetros . . . . .	94
4.6. Pieza principal vista lateral acotada en milímetros . . . . .	95
4.7. Acero 1020-Esfuerzos . . . . .	101
4.8. Aluminio-Esfuerzos . . . . .	101
4.9. ABS-Esfuerzos . . . . .	102
4.10. Acero 1020-Deformación. . . . .	103
4.11. Aluminio- Deformación . . . . .	103
4.12. ABS- Deformación . . . . .	104
5.1. Cinemática <i>Matlab®/Simulink</i> . . . . .	106
5.2. Sistema lazo abierto . . . . .	107
5.3. Trayectorias . . . . .	108
5.4. Respuesta modelo cinemático . . . . .	109
5.5. Controlador proporcional . . . . .	109
5.6. Controlador proporcional con $K_1=2$ y $K_2=2$ . . . . .	110
5.7. Controlador proporcional con $K_1 = 0.8$ y $K_2=0.8$ . . . . .	110
5.8. Resultados no deseados del controlador proporcional . . . . .	112
5.9. Resultados deseados del controlador proporcional . . . . .	112
5.10. Controlador difuso . . . . .	113
5.11. Variables de entrada . . . . .	113
5.12. Variables de salida . . . . .	114
5.13. <i>Surface</i> del controlador difuso . . . . .	116
5.14. Diagrama del controlador difuso . . . . .	116
5.15. Salida controlador difuso . . . . .	117
5.16. Resultado controlador difuso . . . . .	118
5.17. Controlador proporcional-difuso . . . . .	118
5.18. Respuesta difusa con $K_1=1.5$ y $K_2=0.8$ . . . . .	119
5.19. Respuesta difusa $K_1=1.1$ y $K_2=0.8$ . . . . .	119
5.20. Respuesta difusa $K_1=0.8$ y $K_2=0.8$ . . . . .	120
5.21. Respuesta difusa $K_1=0.2$ , $K_2=0.8$ . . . . .	120
5.22. Resultados no deseados del controlador proporcional-difuso . . . . .	122
5.23. Resultados deseados del controlador proporcional-difuso . . . . .	123
5.24. Mejor respuesta variando el ángulo de serpenteo . . . . .	123
6.1. Diagrama de bloques usando « <i>To Workspace</i> » . . . . .	142
6.2. <i>Toolbox</i> de <i>Matlab®/Ident</i> . . . . .	143
6.3. <i>Import data</i> . . . . .	143
6.4. Datos de entrada . . . . .	144

6.5. Identificación . . . . .	144
6.6. Modelado para la función de transferencia . . . . .	145
6.7. Proceso de estimación . . . . .	145
6.8. Resultado de modelo deseado . . . . .	146
6.9. Ganancia y frecuencia de función de transferencia del eje X . . . . .	148
6.10. Ganancia y frecuencia de función de transferencia del eje Y . . . . .	148
6.11. Sistema de control PID . . . . .	149
6.12. Respuesta del sistema con controlador PID . . . . .	149

# Índice de cuadros

1.1. Dimensiones de robot serpiente y rendimiento . . . . .	23
2.1. Regla de sintonía de Ziegler-Nichols del primer método . . . . .	35
2.2. Regla de sintonía de Ziegler-Nichols del segundo método . . . . .	36
2.3. Comparación controladores PID discreto . . . . .	46
2.4. Reglas controlador PID-Fuzzy . . . . .	59
2.5. Comparación de controladores . . . . .	64
3.1. Parámetros Denavit-Hartenberg del robot ápedo. . . . .	76
4.1. Características del servomotor SG-5010 de Tower pro . . . . .	96
4.2. Resultados de materiales . . . . .	105
5.1. Variación de constantes del controlador proporcional . . . . .	111
5.2. Reglas difusas implementadas . . . . .	115
5.3. Variación de constantes del controlador proporcional- difuso . . . . .	121
5.4. Variación del error mínimo en torno al ángulo . . . . .	124
5.5. Comparación de controladores . . . . .	124
5.6. Ventajas y desventajas de los controladores diseñados . . . . .	125



## RESUMEN

La robótica es un mundo extenso y fascinante que se puede clasificar en dos tipos de robots: robots fijos y robots móviles. En la rama de robots móviles se destacan los robots del tipo explorador, los cuales son utilizados para la exploración de zonas de difícil acceso o que presentan algún riesgo para la vida humana, por lo que es necesario hacer un robot que cumpla las necesidades del terreno ocasionando la creación de diferentes diseños que contemplan los distintos parámetros como son: el diseño mecánico, electrónico y de control volviéndose un tema extenso para su estudio. Por esto, el presente trabajo de grado hace énfasis sobre los robots del tipo ápedo enfocándose en el diseño de un controlador.

Los robots ápedos son diseñados para lugares estrechos como son las grietas, túneles y tuberías. Para analizar de forma adecuada se realizó el modelado matemático del movimiento sinusoidal característico del este tipo de robot, junto con el análisis cinemático comprendido por la cinemática directa e inversa. Además, son contempladas las cualidades de locomoción a través de la velocidad angular con el fin de comprender las características que proporciona y tener los fundamentos teóricos para el diseño del controlador.

En los siguientes capítulos a parte del modelado matemático del robot también se presenta el análisis del módulo unitario empleado para componer la estructura principal, realizando el estudio dinámico del módulo y el correspondiente análisis de materiales comúnmente usados para la fabricación de la estructura, para finalmente diseñar un controlador genérico utilizando los controladores PID, difuso y PID-FUZZY con base a la cinemática directa; realizando la correspondiente simulación. Y finalmente, se prueba la aplicación adecuada expuesta en este trabajo monográfico de grado.

**Palabras clave:** Ángulo de serpenteo, cinemática, controlador difuso, controlador PID, controlador PID-FUZZY, modelado matemático, movimiento sinusoidal, robot ápedo.

## ABSTRACT

Robotics is a vast and fascinating world that can be classified into two types of robots: fixed robots and mobile robots. In the branch of mobile robots are highlights the explorer robots,, which are used for the exploration of areas of difficult access or present a risk to human life, so it is necessary to make a robot that meets the needs on the ground in the field of mobile robots causing the creation of different designs that include various parameters such as: mechanical, electronic and control design becoming a vast subject for study. Therefore, this degree work emphasis on legless type robots focusing on the design of controller.

The legless robots are designed for narrow places like cracks, tunnels and pipes. To properly analyze mathematical modelling of sinusoidal motion characteristic of this type of robot was made, along with the kinematic analysis comprised of direct kinematic and inverse kinematic. Moreover, we are referred to the qualities of locomotion through the angular velocity in order to understand the features provided and have the theoretical grounds for controller design.

In the following chapters besides the mathematical modelling of the robot, also it presented the analysis of a unit module used for composing the main structure of robot, performing the dynamic study of the module and the corresponding analysis of materials commonly used for manufacturing the structure, to finally design a generic driver using fuzzy controller and proportional-fuzzy controller based on the kinematic modelling; performing the corresponding simulation. And finally, the proper application exposed in this monograph grade tested.

**Keywords:** Meandering angle, kinematics, fuzzy controller, proportional-fuzzy controller, mathematical modelling, sinusoidal motion, legless robot.

# Capítulo 1

## INTRODUCCIÓN

Con el paso del tiempo, en el mundo se ha visto el desarrollo de diferentes tecnologías con el fin de facilitar la realización de ciertas actividades como la exploración en diferentes lugares que pueden presentar un nivel de dificultad considerable para su exploración o riesgo para la vida humana. La tecnología naciente para suplir esta necesidad es centrada en los robots móviles del tipo explorador diseñados para poder desplazarse en diferentes terrenos y cumplir diversas tareas, debido a esto se encuentra una gran variedad de tipos de robots exploradores con diferentes capacidades y variaciones, donde se destaca el robot del tipo ápedo.

Los robots ápedos se caracterizan principalmente porque no poseen extremidades como piernas y brazos, tienen una estructura angosta ya que son utilizados con fines de exploración en lugares de difícil acceso como grietas, tuberías, entre otros; permitiendo diversos usos en diferentes campos como por ejemplo el presentado por la Nasa que realiza estudios en otros planetas a través de la exploración empleando robots ápedos y en la industria minería es usado en la búsqueda de minerales al poder desplazarse en medio de grietas, entre otros diferentes campos, donde se expone el gran campo de exploración de este tipo de robot tanto mecánico, electrónico como de control .

En este documento se realiza un estudio comparativo de distintas clases de controladores con el objetivo de ser utilizados en un robot ápedo. Adicionalmente, se realiza el modelado matemático del sistema con el fin de analizar el movimiento característico del robot ápedo y posteriormente proponer un sistema que sea aplicable en los robots exploradores tipo ápedo que permita controlar la trayectoria del robot haciendo énfasis en el módulo principal de la estructura.

### 1.1. Planteamiento y formulación del problema

En la actualidad encontramos gran variedad de robots móviles entre los que se destacan los robot exploradores que surgen de la necesidad constante de buscar e investigar

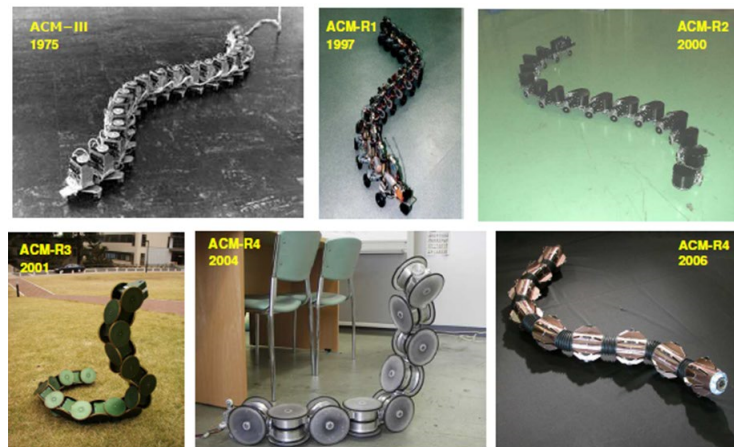
en lugares donde el ser humano no puede acceder, ya sea por el peligro o la dificultad de desplazamiento. Uno de los robots exploradores más destacados es el robot ápodo porque es capaz de adaptarse a diferentes entornos y movilizarse en lugares estrechos, por lo que es utilizado normalmente en la exploración de tuberías como el robot «*Snake Robot*»[29] o el robot robot «*USAR*»[10] diseñado para rescate.

Al ser utilizados en diferentes campos, los robots ápodos son diseñados para tareas específicas por lo que tanto el sistema mecánico como el sistema de control varían según las necesidades del usuario.

### 1.1.1. Descripción del problema

Al profundizar en el área de robots exploradores del tipo ápodo, se encuentran una gran variedad de diseños que al paso del tiempo han ido mejorando, buscando una mejor funcionalidad como el que se muestra en la (*Figura 1.1*). Al mostrar los cambios que ha obtenido el modelo de los robots serpiente de la familia ACM (*Active Cord Mechanism*) en que va cambiando su estructura para mejorar su diseño y por lo tanto su sistema de control.

Figura 1.1: Evolución de los robots serpiente de la familia ACM (Active Cord Mechanism) Hirose- Fukushima Robotics Lab



Fuente: Juan González Gómez [11]

Como el ejemplo propuesto en el párrafo anterior usando como referencia el robot serpiente ACM (*Figura 1.1*) se encuentra una gran variedad de diseños, los cuales cambian según el enfoque de cada uno y por ende manejan un sistema de control distinto ocasionando que los ingenieros no se enfoquen en el diseño mecánico sino en un sistema de control adecuado que cumpla las funciones requeridas por el robot a pesar que se basan en el movimiento sinusoidal característico de la serpiente.

### 1.1.2. Formulación del problema

Conociendo los retos de diseñar un robot ápedo tanto en los aspectos mecánicos como en el sistema de control y conociendo las cualidades mecánicas al basarse en el movimiento sinusoidal, ¿Cómo se podría diseñar un sistema de control adecuado para los robots ápedos considerando los parámetros que se deben tener en cuenta para su correcto funcionamiento?

## 1.2. Justificación

Los robots exploradores pertenecen a la familia de la robótica móvil y tienen como característica principal que están diseñados especialmente para el desplazamiento en lugares peligrosos o de difícil acceso para el ser humano. Para lograr esto es necesario considerar el entorno en el que el prototipo se irá a desplazar, los cambios que éste puede sufrir en el trayecto y buscar una estrategia de control que puede tener según las necesidades de su estructura, por lo que normalmente se utilizan robots basados en animales o más conocidos como robots zoomorfos<sup>1</sup>, debido a que los animales han evolucionado para adaptarse a entornos específicos. Uno de los robots zoomorfos que poseen un campo extenso de investigación pero poco común a pesar de su sencillez, son los ápedos mejor conocidos como robot serpiente o *snake robot* (en inglés), es diseñado para zonas estrechas y de difícil acceso como derrumbes, industria minera o de gases, por lo tanto debe contar con la capacidad de adaptarse y desplazarse en sitios muy angostos, lo cual presenta un reto en el diseño y control, pero con gran utilidad a nivel industrial o de rescate.

Al ser un reto tanto en el diseño mecánico como en sistema de control se deben observar las cualidades que poseen en estos dos aspectos. En los aspectos mecánicos de los robots ápedos se destacan los diseños de movimiento sinusoidal característico de este robot al ser un movimiento común entre los animales ápedos, pero el sistema de control cambia gracias a la utilidad que presenta el robot, debido al cambio de funcionalidad de este tipo de robot se requieren diferentes clases de sensores como también puede tener diferentes cualidades, por lo que varía el sistema de control acorde a esto.

## 1.3. Estado del arte

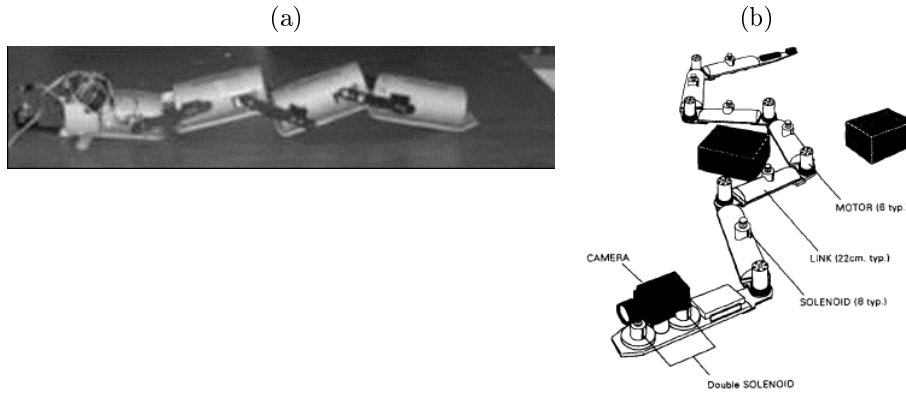
La investigación de la robótica móvil entiende una gran variedad de robots diseñados para la exploración en distintos terrenos y su adaptación frente a los cambios externos. Entre los robots que cumplen esas cualidades se destaca el robot ápedo, dada su característica de adaptarse mejor a la variación de terreno gracias a su movimiento sinusoidal, un ejemplo de la implementación del movimiento de los robots ápedos con

---

<sup>1</sup>Los robots zoomorfos son aquellos que emplean un sistema de locomoción similar al de los animales.

objetivos de investigación es presentado por Méndez J.et.al. [21] con base a un sistema mecánico en dos dimensiones (2D) mostrado en la (Figura 1.2a). Otro campo de investigación fue expuesto por Yansong Shan y Yoram Koren [25] en el año de 1993, implementando un sistema de movimiento controlado por cámara posicionada en la parte frontal del robot como se ilustró en la (Figura 1.2b).

Figura 1.2: Robots ápodos con movimiento en 2D



Fuente: 1.2a Mendez J.[21], Sistema mecánico en 2D; 1.2b Yansong Shan, Yoram Koren [25], Robot de forma cilíndrico con desplazamiento en 2D.

Los robots ápodos han sido desarrollados en distintos campos expuestos por Juan González Gómez [10], mostrando distintas líneas de investigación que van desde inspección ya sea de puentes al revisar la soldadura en los puntos de conexión como la búsqueda entre escombros para rescate como se muestra en la (Figura 1.3) implementando el robot USAR (*Urban Search and Rescue*) hasta el robot anfibio presentado por Lipkin K.et.al.[17] en «*Conference on Intelligent Robots and Systems*» de 2007.

Figura 1.3: Robot USAR para rescate



Fuente: Kalani H.et.al. [10]

El enfoque de robot ápedo del tipo anfibio fue tratado de igual manera por el trabajo de Crespi A.et.al. [6] presentando en el 2005, empleó un robot serpiente mostrado en la (*Figura 1.4a*) con las capacidades de escalar y nadar dirigido a la implementación en zonas pantanosas, ese mismo trabajo fue retomado en el año 2008 por Alessandro Crespi y Auke Jan Ijspeert [22] miembros investigadores del anterior trabajo exponiendo un robot mejorado mostrado en la (*Figura 1.4b*), el cual implementó un sistema de control neuronal mejorando la respuesta del sistema mecánico del robot.

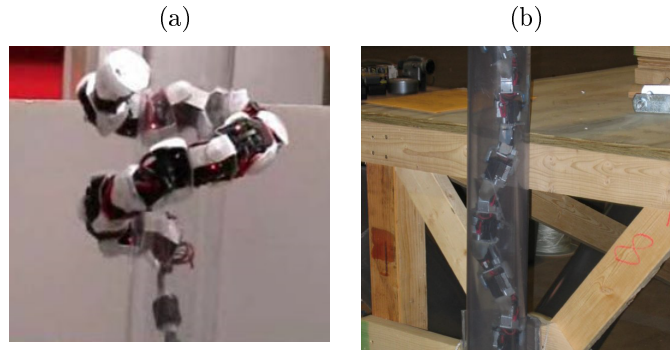
Figura 1.4: Robots ápodos anfibios



Fuente: 1.4a Ro Crespi, Badertscher A.et.al.[6], Robot con ruedas pasivas, 1.4bAuke Jan Ijspeert Ro Crespi [22], Robot con seguidor LED.

En la industria, María Alejandra Urdaneta y Roque Saltaren [27] presentaron modelos de robots ápodos con sistemas de locomoción adaptable a tuberías, con el enfoque en el campo de la extracción petrolera con bombas BES (*bombas electro sumergibles*) y BCP (*bombas de cavidad progresiva*) en la inspección interna de tuberías para búsqueda de fugas o imperfectos en el sistema de flujo de petróleo. Pero Wright C.et.al. [29] propusieron un robot modular expuesto en «*Conference on Intelligent Robots and Systems*» año 2007 con capacidad de explorar en tuberías tanto externamente como se puede ver en la (*Figura 1.5a*) e internamente mostrado en la (*Figura 1.5b*) gracias al gran número de articulaciones que proporciona un significativo número de grados de libertad. Ese robot fue empleado por David Rollinson y Howie Choset [7] como punto de referencia en su trabajo buscando definir el movimiento de los robots ápodos desde la perspectiva externa del robot debido a los cambios constantes realizados internamente por la variabilidad de la locomoción interna, siendo analizado de forma virtual para separar los cambios internos con los externos facilitando el análisis de ese robot, ese trabajo fue presentado en «*International Conference on Intelligent Robots and Systems*» año 2011.

Figura 1.5: Robots ápodos escaladores



Fuente: Wright C.et.al.[29], 1.5a Robot serpiente trepador, 1.5b Robot ápodo escalando el interior de un tubo.

La idea de robots con múltiples grados de libertad nace a partir de los robots hiper-redundantes. Según Samuel Muñoz González [18] en su trabajo de maestría del año 2013 mostró a los robots ápodos como robots hiper-redundantes bio-inspirados para lograr una clase de robot móvil con el fin de imitar los movimientos de animales tanto vertebrados como invertebrados, desplazándose en lugares de difícil acceso. El proyecto buscó minimizar los errores a través de algoritmos. Uno de los robots usados como objeto de estudio fue el robot ápodo «Uncle Sam» mostrado en la (Figura 1.6) diseñado para realizar movimientos suaves y complejos, se desplaza entre grietas y escombros para localizar víctimas atrapadas gracias al equipamiento de una cámara en la parte frontal del cuerpo.

Figura 1.6: Uncle Sam



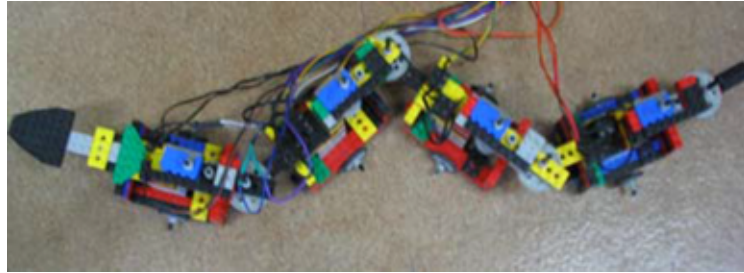
Fuente: Samuel Muñoz González [18]

Entre la variabilidad de diseños se encuentra el propuesto por Laxmisha Rai y Soon Ju Kang [16] mostrando un robot serpiente compuesto de figuras lego junto con los respectivos motores y sensores como se muestra en la (Figura 1.7), el cual fue utilizado para realizar el análisis matemático del movimiento sinusoidal que les permitió plantear un sistema de control para sí mismo y elaborar el respectivo análisis. Otro



sistema de estudio fue el propuesto por Andrés San Millán Rodríguez [24] haciendo el estudio de la locomoción sinusoidal empleada como base para el desarrollo de un robot de elaboración propia con locomoción utilizando servomotores, informando de sus cualidades electrolíticas requeridas por su modelo y finalizando con una simulación utilizando la herramienta V-REP en la que el robot modular realizó las mismas tareas tanto en el software como físicamente.

Figura 1.7: Robot ápodo LEGO



Fuente: Laxmisha Rai y Soon Ju Kang [16]

Asimismo, Tully S.et.al. [26] propusieron otro tipo de robot con su estructura flexible caracterizado por tener dos partes principales, también llamadas serpientes interna y externa, que cumplen con la función de mejorar el desplazamiento y así poder lograr una mejor locomoción. Los robots de estructura flexible son importantes porque ayudan a realizar estudios y diseños en el campo de la medicina, más específicamente, para cumplir la función de realizar cirugías especialmente en el corazón, por lo que se crea un algoritmo de estimación para lograr una visualización en 3D, lo que representa una forma precisa de lograr una cirugía guiada por imagen. Otro tipo de sistema son los micro-controladores utilizados en el robot AmphiBot [6], el robot serpiente (*Snake Robot*) [29] o el robot estudiado por Gonzalez J.et.al. [13] que compara el tiempo de respuesta de varios controladores respecto a un robot diseñado con base de “*Cube Revolutions*” como se muestra en la (Figura 1.8).

Figura 1.8: Cubos en revolución



Fuente: González J.et.al.[13]

Para lograr esto, es necesario realizar el análisis matemático correspondiente como se muestra Vossoughi G.et.al. [28] al analizar los ángulos de giro de la estructura y las fuerzas que ejercen para lograr el movimiento sinusoidal característico de los robots ápodos debido a que se mueve formando una onda sinusoidal con base a la dirección deseada para la locomoción. A pesar de ser un movimiento en dos dimensiones puede usarse para el desplazamiento tridimensional según el diseño del robot y el plano en que se esté desplazando. Los resultados de estos movimientos pueden verse en la simulación de forma más clara y real al contemplar mejor los factores del diseño como se exponen Kalani H.et.al. [15] que evalúa el sistema con base al diseño en la simulación y los datos resultantes de este análisis, junto con las posibles variaciones que podría tener. Uno de los factores más importantes y que también contempla a la anterior cita es el gasto de energía según la distancia recorrida y las posibles variaciones que podrá tener en el desplazamiento y el correspondiente análisis debido a la necesidad de cargar celdas de energía al ser un robot móvil.

En el estudio realizado por Kalani H.et.al. [15] en su trabajo «*Application of statistical techniques in modeling and optimization of a snake robot*» presenta un estudio en el que compara múltiples robots ápodos presentados en el cuadro 1.1, resaltando las dimensiones de los módulos, número de módulos y longitud total exhibiendo así las características de cada uno.

Cuadro 1.1: Dimensiones de robot serpiente y rendimiento

Robot	Longitud de cada eslabón	Longitud total	Sección transversal	Peso total	Velocidad	No. de enlaces o módulos
ACM III	144	2000	0.023	28	400	20
AmphiBot I (AB I)	70	490	0.002	–	35	8
AmphiBot II (AB II)	94	770	0.002	–	400	8
KR-II	–	3300	0.497	370	500	7
KR-I	–	1390	0.081	27.8	266	6
OmniTread (OT-8)	200	1270	0.034	13.6	100	5
OmniTread (OT-4)	–	940	0.007	3.6	150	7
JL-I	350	1050	0.038	21	180	3
Kotay's Inchworm I (KIR-1)	–	250	–	0.455	4	–
Kotay's Inchworm II (KIR-2)	–	330	–	0.566	13	–
CMU (M1)	–	840	0.003	1.26	102	–
Slim Slime Robot (SSR)	177.6	730	0.013	12	60	6
Planar Inchworm (PI)	–	710	–	6	1	2
FUM Snake-2	100	1600	0.48	4.48	150	16

Fuente: Kalani H.et.al.[15]

## 1.4. Objetivos

### 1.4.1. General

- Diseñar un sistema de control genérico para robot exploradores tipo ápodos que permita controlar múltiples celdas unitarias.

### 1.4.2. Específico

- Formular el modelado matemático del sistema con el fin de analizar el movimiento característico de los robots ápodos.
- Fijar los parámetros característicos del robot ápodo (Material básicos de los robots ápodos y limitaciones mecánicas).
- Realizar un estudio comparativo con las distintas clases de controladores.
- Mostrar los resultados a través de simulaciones resaltando las diferencias entre los distintos controladores.

## 1.5. Alcances y limitaciones

El proyecto de grado se desarrolla a través de una conexión entre la interfaz gráfica *MATLAB*<sup>®</sup> donde se realiza toda la programación del control para el robot ápodo simulando el comportamiento de robots pre-diseñados. La sinergia de las herramientas anteriormente mencionadas permiten desarrollar un sistema que simulación logrando realizar un estudio conciso de cómo funciona un robot ápodo, permitiendo su análisis matemático y su implementación.

La limitación del proyecto se refleja en la falta de implementación física debido al costo del robot y falta de recursos, por esto, se recurre a una plataforma simulada donde se van a obtener los mismos resultados.

## 1.6. Línea de investigación de programa

La línea de investigación en la que se dirigirá el proyecto es:

- Robótica y biomecatrónica.

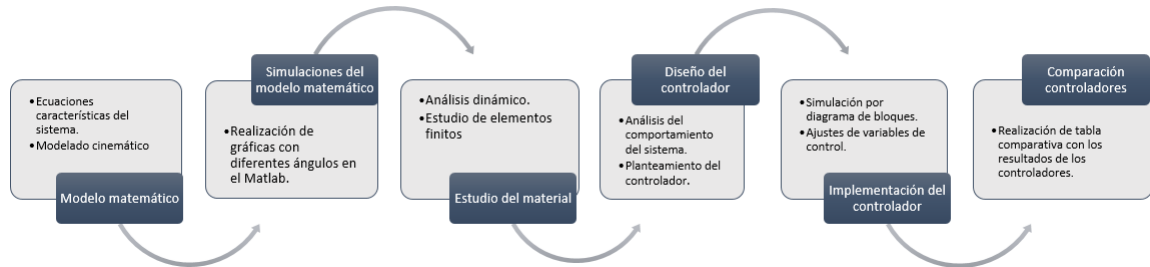
El proyecto consiste en el diseño de un controlador para los robot tipo ápodo con el objetivo de disminuir el error de la trayectoria y adaptarse tanto a las necesidades del usuario como las de sí mismo para facilitar su control, por lo que se desempeñará en el área de control de la robótica. Por lo tanto la línea de investigación del proyecto es la mencionada anteriormente.

## 1.7. Diseño metodológico

La metodología utilizada en este proyecto se basó en una serie de procesos que se pueden observar en la (*Figura 1.9*) y con los que sirvieron como herramienta para el cumplimiento de todos los objetivos planteados.

Adicionalmente, permitió dar solución a ciertas dudas e inconvenientes presentados en el desarrollo del trabajo.

Figura 1.9: Metodología utilizada



Fuente: Elaboración propia

## 1.8. Contribuciones

Dentro del desarrollo del presente trabajo, se logró durante el desarrollo del proyecto publicar un artículo, el cual se expone en el anexo A.

- Jonathan A. Bermúdez M., Claudia T. Martínez S., Rubén D. Hernández B., Oscar Avilés S. Diseño y construcción de un robot gusano tipo explorador. En: **Segunda Conferencia Internacional de Mecatrónica avanzadas, Diseño y Tecnologías de Fabricación - AMDM 2014.**

También, se tuvo participación:

- Jonathan A. Bermúdez M., Claudia T. Martínez S., Rubén D. Hernández B. Robot explorador gusano. En: **Encuentro de robótica y bio-mecatrónica en la Universidad Jorge Tadeo Lozano 2014.**

Además, se sometió el artículo visto en el anexo B.

- Jonathan A. Bermúdez M., Claudia T. Martínez S., Rubén D. Hernández B., João M. Rosario., Oscar Avilés S. Análisis cinemático de un robot ápedo. En: **V Congreso Internacional de Ingeniería Mecatrónica y Automatización - CIIMA Bucaramanga 2016.**

## Capítulo 2

# MARCO TEÓRICO

A lo largo de tiempo se han creado una gran cantidad de tipos de robots móviles que han ayudado en distintas áreas de trabajo entre las que se puede destacar a los robots exploradores, debido a que han facilitado la exploración de lugares peligrosos para ser humano o de difícil acceso, debido a esto se ha necesitado crear robots que se puedan adaptar cada vez mejor al área de exploración, dando como resultado los robots bio-inspirados<sup>1</sup> entre los que destacan los robots ápodos, en el que está enfocado este trabajo.

Inspirado en la naturaleza los ápodos se centran en el movimiento sinusoidal que permite su desplazamiento, característica de los animales e insectos que no poseen extremidades como son los gusanos y las serpientes. Este tipo de movimiento presenta una gran variedad de cualidades entre las que se resaltan las siguientes:

- Requiere un espacio mínimo para su desplazamiento por lo que puede desplazarse entre agujeros, escombros y grietas.
- La facilidad con la que se puede adaptar al entorno al basarse en celdas unitarias.
- Pueden desplazarse por diferentes terrenos.

Las anteriores cualidades se pueden ver claramente en [6] que muestra un robot ápodo de tipo anfibio mostrado en la (*Figura 2.1a*), construido con componentes en poliuretano<sup>2</sup> aligerado con micro-esferas de fenol<sup>3</sup> para asegurar la impermeabilidad con celdas rectangulares que facilita su desplazamiento sobre el agua como también posee ruedas para su desplazamiento en tierra. Al realizar su desplazamiento utiliza motores DC controlados por un micro-controlador (*PIC*) lo cual generó un gran avance permitiendo desarrollar el robot ápodo de tipo anfibio optimizado [22] mostrado en la (*Figura 2.1b*).

---

<sup>1</sup>Los robots bio-inspirados son aquellos que se basan en la naturaleza.

<sup>2</sup>Poliuretano es un polímero plástico ligero que puede ser rígido o flexible. Posee una gran cantidad de aplicaciones gracias a sus cualidades.

<sup>3</sup>El fenol en su forma pura es un sólido cristalino de color blanco-incoloro a temperatura ambiente soluble a algunos químicos.

El robot busca mejorar su sistema de control empleando un sistema de redes neuronales utilizando un sistema GPS mejorando su control para sus tareas terrestres como acuáticas.

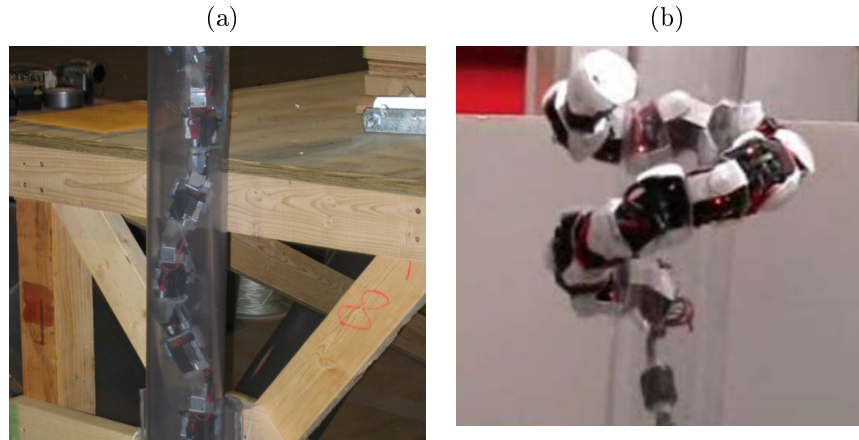
Figura 2.1: Robots ápodos anfibios



Fuente: 2.1a Ro Crespi, Badertscher A.et.al.[6], Robot con ruedas pasivas; 2.1b Auke Jan Ijspeert Ro Crespi [22], Robot con seguidor LED.

Otra de las cualidades de los robots ápodos se puede ver más claramente en [29] que usa 16 módulos capaces de realizar funciones como trepar por el interior de un tubo gracias a la utilización de servomotores como actuadores al proporcionar un grado de libertad por articulación ocasionando que tenga que dividir en dos grupos que proporcionan grados verticales como horizontales logrando un movimiento en 3D, compuesto como se muestra en la (Figura 2.2a y 2.2b). El sistema de control usado para el robot mostrado se empleó un microprocesador Atmel permitiendo controlar el pulso de PWM necesario para controlar el ángulo para los servomotores.

Figura 2.2: Robots ápodos trepadores



Fuente: Wright C.et.al. [29], 2.2a Robot serpiente escalador, 2.2b Robot serpiente trepador.

Observando las cualidades necesarias para los robots redundantes es inevitable hablar de los robots hiper-redundantes los cuales emplean gran cantidad de grados de

libertad girando sobre su centro como se muestra en la (*Figura 2.3*), la cual mantiene un punto fijo permitiendo obtener una mejor precisión usando módulos en forma de disco, lo que permite mayor flexibilidad. Utilizando ese sistema mecánico que se mueve a través de servomotores se implementó la tarjeta arduino como sistema de control permitiendo sincronizar las distintas partes del robot. Este mecanismo también es uno de los implementados en los robots ápodos debido a la necesidad de control sincrónico de las múltiples celdas unitarias que poseen los robots ápodos del tipo escalador.

Figura 2.3: Robot hiper-redundante trompa de elefante



Fuente: Samuel Muñoz González [18]

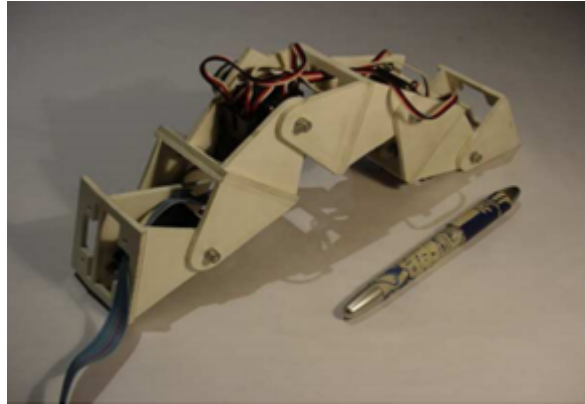
Teniendo en cuenta los distintos diseños mostrados anteriormente se encuentra una gran diferencia entre los tipos de controladores utilizados debido a que se debieron adaptar a las necesidades mecánicas del sistema, gracias a esto surge la necesidad de mejorar los controladores para los distintos cambios que se obtienen a pesar de basarse todos en el movimiento sinusoidal característico de los robots ápodos, gracias a esa problemática se han planteado múltiples trabajos como [12] que presenta una unidad de control basada en FPGA (por sus siglas en inglés «*Field Programmable Gate Array*»)<sup>4</sup>, usando como base el robot ápodo «*Cube Reloaded*» visto en la (*Figura 2.4*) permitiendo el análisis del movimiento en 2D mejorando la visualización del sistema de locomoción.

---

<sup>4</sup>FPGA es un semi-conductor que contiene bloques de lógica programables.



Figura 2.4: El robot ápedo Cube Reloaded



Fuente: González J.et.al. [12]

Como también se han comparado sistemas de control implementado como en [13], que usa de punto de referencia el desempeño del robot ápedo «*Cube Reloaded*» mencionado anteriormente. En ese trabajo evaluaron varios procesadores embebidos en FPGA y midiendo la respuesta en distintos aspectos para compararlos entre sí, evaluando el desempeño del controlador frente al movimiento sinusoidal característico del robot y midiendo el tiempo de ejecución del algoritmo.

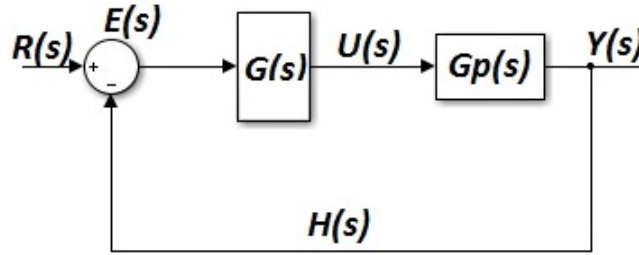
## 2.1. Controladores

Los controladores son una parte fundamental para obtener las respuestas deseadas de un determinado sistema de acuerdo a los diferentes parámetros establecidos por el usuario. Entre los controladores comúnmente usados se encuentran el controlador PID (Proporcional Integrador Derivador), difuso y difuso-PID los cuales son presentados a continuación.

### 2.1.1. PID (proporcional integrativo derivativo)

El controlador PID (Proporcional Integrativo Derivativo) es una herramienta de control genérico utilizado en sistemas de lazo cerrado. El controlador PID tiene en cuenta el error obtenido de la salida resultante del sistema menos la entrada como es expuesto en el diagrama de la (Figura 2.5) donde la entrada es denotada como  $R(s)$ , el error es denotado como  $E(s)$ , el controlador PID se denota como  $G(s)$ , la salida del controlador a la planta es denotada como  $U(s)$ , el modelo matemático de la planta es denotada como  $Gp(s)$ , la retroalimentación es denotada como  $H(s)$  y la salida del sistema es denotado como  $Y(s)$ .

Figura 2.5: Control PID



Fuente: Elaboración propia

Como se muestra en el libro «*Sistemas de control moderno*» de Richard Dorf [8] el controlador PID está compuesto por tres componentes los cuales son:

- **Acción proporcional (P):** La acción proporcional es el resultado del producto entre la ganancia proporcional  $K_p$  y el error en estado estacionario  $E(t)$ ; el resultado del producto dado en la ecuación 2.1 que es igual a la ecuación 2.2 para realizar la función de transferencia, que alcanza su estado estacionario cuando el resultado del producto es igual al resultado del error estacionario.

$$P = K_p * E(t) \quad (2.1)$$

$$P = K_p * E(s) \quad (2.2)$$

- **Acción integral (I):** La acción integral se obtiene como resultado de la multiplicación entre la ganancia integral  $K_i$  y la integral definida del error estacionario  $E(t)$  como se muestra en la ecuación 2.3 o para poder hacer la ecuación en función de transferencia 2.4. Al evaluarse la integral del error estacionario se tienen en cuenta los resultados anteriores del error ocasionado que sólo alcance su valor estacionario cuando el error sea igual a cero.

$$I = K_i * \int_0^t E(t) dt \quad (2.3)$$

$$I = \frac{K_i}{s} * E(s) \quad (2.4)$$

- **Acción derivativo (D):** La acción derivativa es el resultado del producto entre la ganancia  $K_d$  y la derivada del error estacionario  $E(t)$  como se muestra en la ecuación 2.5 y en la ecuación 2.6 para hacer la función de transferencia. Debido a que emplea la derivada del error estacionario tiene una respuesta anticipada por lo que genera una reacción instantánea, por lo que se vuelve nulo cuando el error adquiere su valor estacionario.

$$D = K_d * \frac{dE(t)}{dt} \quad (2.5)$$

$$D = K_d * s * E(s) \quad (2.6)$$

- **Acción PID:** La acción PID es lograda cuando se combinan las tres acciones mencionadas anteriormente dando como resultado la salida  $U(s)$ ; las ecuaciones correspondientes a la suma de los componentes del controlador dan como resultado la ecuación 2.7 y empleándola para realizar la función de transferencia se obtiene la ecuación 2.8.

$$U(t) = K_p * E(t) + K_i * \int_0^t E(t) dt + K_d * \frac{dE(t)}{dt} \quad (2.7)$$

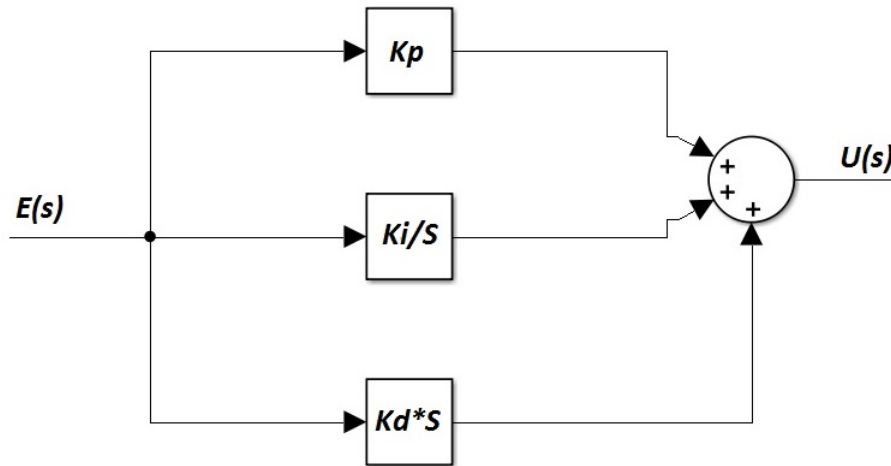
$$U(s) = \left( K_p + \frac{K_i}{s} + K_d * S \right) * E(s) \quad (2.8)$$

Obteniendo la función de transferencia utilizando la ecuación 2.8 es obtenida la ecuación 2.9.

$$G(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d * S \quad (2.9)$$

El bloque de PID visto en la (Figura 2.5) denotado como  $G(s)$  se puede re-escribir como la (Figura 2.6) usando la acción PID.

Figura 2.6: Diagrama de PID



Fuente: Elaboración propia

Como es exhibido en el libro «Ingeniería de control moderna» de Katsuhiko Ogata [19] se muestran tres métodos de diseño de controlador PID que consideran diferentes parámetros de diseño para su desarrollo como se muestra a continuación.

**2.1.1.1. Respuesta en frecuencia**

La respuesta en frecuencia es un método de diseño que analiza la respuesta del sistema en estado estable teniendo en cuenta algunos parámetros de diseño como son: el error estático en la velocidad, el margen de fase permitido y el margen de ganancia. Un ejemplo del método de respuesta en frecuencia es el mostrado en el libro de Katsuhiko Ogata [19] que emplea la planta  $Gp(s)$  descrita en la ecuación 2.10.

$$Gp(s) = \frac{1}{s^2 + 1} \quad (2.10)$$

Es seleccionado el controlador PID ( $G(s)$ ) adecuado para la planta que tiene la forma expuesta en la ecuación 2.11.

$$G(s) = \frac{K(As + 1)(Bs + 1)}{s} \quad (2.11)$$

Después de finalizar la selección de parámetros con los que se busca hallar el valor de las incógnitas del controlador seleccionado es decir K, A y B. La primer incógnita a encontrar es K la cual es posible encontrar con base al error estático  $K_v$  que en este caso se seleccionó que fuera  $4\text{seg}^{-1}$  por lo que al ser planteado es encontrada la ecuación 2.12.

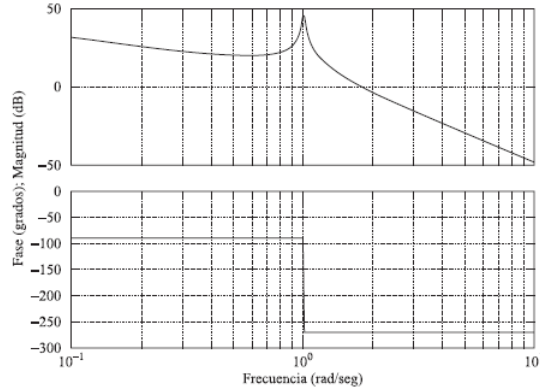
$$K_v = \lim_{s \rightarrow 0} sK \frac{(As + 1)(Bs + 1)}{s} * \frac{1}{s^2 + 1} = K = 4 \quad (2.12)$$

Debido a esto la ecuación 2.11 es re-escrita reemplazando la variable K por su valor que en este caso es 4 obteniendo la ecuación 2.13.

$$G(s) = \frac{4(As + 1)(Bs + 1)}{s} \quad (2.13)$$

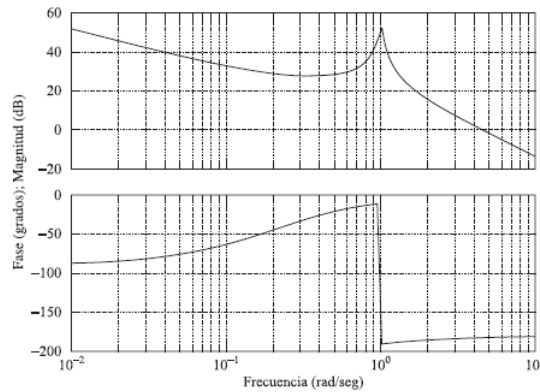
Considerando que otro parámetro de diseño es que el margen de fase sea menor a  $50^\circ$ , el margen de ganancia 10 dB y considerando que las variables para esta parte sean iguales a cero se encuentra que el resultante del producto entre la planta y el controlador sería descrito en la ecuación 2.14.

$$N(s) = \frac{4}{s(s^2 + 1)} \quad (2.14)$$

Figura 2.7: Diagrama de *Bode* para  $N(s) = \frac{4}{s(s^2+1)}$ 

Fuente: Katsuhiko Ogata [19]

Al realizar el diagrama de «*Bode*» (visto en la *Figura 2.7*) que usa como referencia la ecuación 2.14 y considerando los parámetros de diseño se encuentra que  $A = 5$  para que cumpla las condiciones que el margen de fase sea menor a  $50^\circ$  y el margen de ganancia 10 dB. Al haber encontrado la incógnita ocasiona que sólo reste hallar  $B$ , que se encuentra con el mismo método a excepción que se considera el nuevo valor de  $A$ , por lo tanto la nueva ecuación usada en el «*Bode*» (mostrada en la *Figura 2.8*) está dada en la ecuación 2.15.

Figura 2.8: Diagrama de *Bode* para  $N(s) = \frac{4(5s+1)}{s(s^2+1)}$ 

Fuente: Katsuhiko Ogata [19]

$$N(s) = \frac{4(5s+1)}{s(s^2+1)} \quad (2.15)$$

Al realizar esto se encuentra que  $B = 0,25$  por lo que la ecuación final del PID está dada en la ecuación 2.16.

$$G(s) = \frac{4(5s+1)(0,25s+1)}{s} \quad (2.16)$$

### 2.1.1.2. Reglas de Ziegler-Nichols para PID

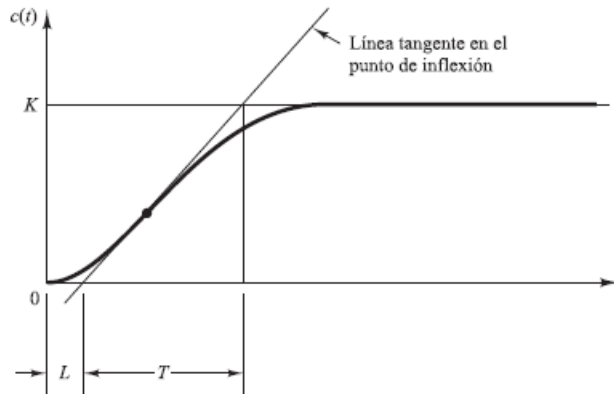
Las reglas para sintonizar controladores PID propuestas por Ziegler y Nichols gira entorno a la ecuación 2.17 cuyas variables corresponden a la ganancia proporcional  $Kp$ , tiempo integral  $Ti$  y tiempo derivativo  $Td$  fundamentado en las características de la respuesta transitoria del sistema, como es expuesto en el libro de «*Ingeniería de control moderna*» de Katsuhiko Ogata [19] donde explica los dos métodos denominados reglas de sintonía de Ziegler-Nichols.

$$PID = Kp \left( 1 + \frac{1}{Ti * S} + Td * S \right) \quad (2.17)$$

#### ■ Primer método de sintonía de Ziegler-Nichols

El primer método es fundamentado en la respuesta del sistema frente a un estímulo del tipo escalón obteniendo los parámetros de forma experimental. Cuando la planta no contiene integradores ni polos dominantes complejos conjugados, la respuesta de la planta causada por el estímulo proporcionado por un escalón unitario puede obtener una forma de S, como se observa en la (Figura 2.9) demostrando que este método sólo se puede aplicar cuando la respuesta obtenida posee una forma de S.

Figura 2.9: Forma de la respuesta en forma de S



Fuente: Katsuhiko Ogata [19]

La curva en forma de S vista en la (Figura 2.9) expone los parámetros de las constantes del tiempo  $T$  y el tiempo de retardo  $L$ , tenidos en cuenta para el desarrollo del primer método. Estos parámetros son determinados mediante el dibujo de una recta tangente al punto de inflexión encontrado en la respuesta en forma de S. La cual corta en la intersección con el eje del tiempo y la línea dada por  $C(t) = K$ .

Obteniendo los valores de las constantes del tiempo  $T$  y el tiempo de retardo  $L$  es utilizado el cuadro 2.1, donde indica las reglas para encontrar los valores de las

constantes de control para un controlador proporcional P, controlador proporcional integrador PI y controlador proporcional, integrador y derivador PID.

Cuadro 2.1: Regla de sintonía de Ziegler-Nichols del primer método

Tipo de controlador	$K_p$	$T_i$	$T_d$
P	$\frac{T}{L}$	$\infty$	0
PI	$0,9 * \frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 * \frac{T}{L}$	$2 * L$	$0,5 * L$

Fuente: Katsuhiko Ogata [19]

Para realizar un ejemplo de la aplicación del primer método se considera una constante de tiempo  $T$  igual a 13 y un tiempo de retardo  $L$  de 5 por lo que utilizando el cuadro 2.1 se encuentran los valores de las variables de la ecuación 2.17 correspondiente al controlador PID dando como resultado:

$$K_p = 1,2 * \frac{T}{L} = 1,2 * \frac{13}{5} = 3,12 \quad (2.18)$$

$$T_i = 2 * L = 2 * 5 = 10 \quad (2.19)$$

$$T_d = 0,5 * L = 0,5 * 5 = 2,5 \quad (2.20)$$

Reemplazando en la ecuación 2.17 las variables obtenidas es encontrada la ecuación 2.21.

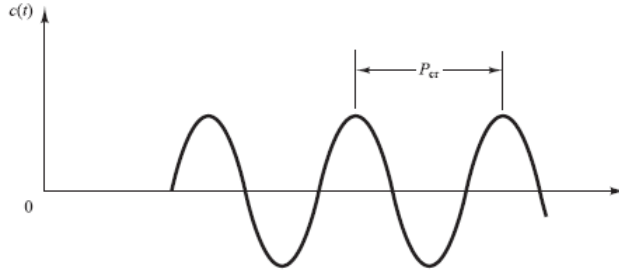
$$PID = 3,12 \left( 1 + \frac{1}{10 * S} + 2,5 * S \right) \quad (2.21)$$

Simplificando la ecuación 2.21 con el fin de obtener la forma convencional del controlador se encuentra la ecuación 2.22.

$$PID = \frac{0,312 + 3,12 * S + 7,8 * S^2}{S} \quad (2.22)$$

### ■ Segundo método de sintonía de Ziegler-Nichols

En el segundo método de sintonía de Ziegler-Nichols son fijados algunos parámetros como  $T_i = \infty$  y  $T_d = 0$  con el fin de poder plantear el controlador con sólo la acción del control proporcional  $K_p$ . Con ese objetivo es incrementado  $K_p$  desde 0 hasta un valor crítico simbolizado con  $K_{cr}$ , en el cual presenta una oscilación sostenida. Otra de las variables consideradas es el periodo  $P_{cr}$  encontrado de forma experimental como se observa en la (Figura 2.10) con el fin de encontrar los valores de las variables  $K_p$ ,  $T_i$  y  $T_d$  de la ecuación 2.17.

Figura 2.10: Oscilación sostenida con periodo  $P_{cr}$  en segundos

Fuente: Katsuhiko Ogata [19]

El cuadro 2.2 son expuestas las reglas para encontrar los valores de las variables para diferentes tipos de controladores como es el controlador proporcional P, controlador proporcional integral PI y controlador proporcional, integral y derivador comprendidos por las variables  $K_p$ ,  $T_i$  y  $T_d$  correspondientes a la ecuación 2.17.

Cuadro 2.2: Regla de sintonía de Ziegler-Nichols del segundo método

Tipo de controlador	$K_p$	$T_i$	$T_d$
P	$0,5 * K_{cr}$	$\infty$	0
PI	$0,45 * K_{cr}$	$\frac{1}{1,2} * P_{cr}$	0
PID	$0,6 * K_{cr}$	$0,5 * P_{cr}$	$0,125 * P_{cr}$

Fuente: Katsuhiko Ogata [19]

Un ejemplo de la aplicación del segundo método es el mostrado por Katsuhiko Ogata [19] donde se plantea una planta mostrada en la ecuación 2.23 y un controlador  $K_p$  dando como resultado la función de salida del sistema mostrada en la ecuación 2.24.

$$Gp(s) = \frac{1}{s(s+1)(s+5)} \quad (2.23)$$

$$Gf(s) = \frac{K_p}{s(s+1)(s+5) + K_p} \quad (2.24)$$

Se sugiere utilizar el criterio de estabilidad de Routh por lo que se emplea la ecuación 2.25 correspondiente al denominador de la ecuación 2.24.

$$Df = s^3 + 6s^2 + 5s + K_p \quad (2.25)$$

Usando el criterio de estabilidad de Routh:

$$\begin{array}{ccc} s^3 & 1 & 5 \\ s^2 & 6 & K_p \\ s & \frac{30-K_p}{6} & 0 \\ s^0 & K_p & \end{array}$$



Como el criterio de estabilidad de Routh dicta que si no hay un cambio de signo se encuentra que  $K_p$  debe ser mayor de 0 y menor que 30 como es descrito en la ecuación 2.26, dando como valor crítico  $K_{cr} = 30$ .

$$0 < K_p < 30 \quad (2.26)$$

Reemplazando la variable  $K_p$  de la ecuación 2.25 con el valor crítico de  $K_{cr} = 30$ , obteniendo la ecuación 2.27.

$$Df = s^3 + 6s^2 + 5s + 30 \quad (2.27)$$

Para obtener la frecuencia de oscilación sostenida, es necesario sustituir  $s$  por el valor en frecuencia  $j\omega$  en la ecuación 2.27 obteniendo como resultado la ecuación 2.28 e igualando a 0.

$$(j\omega)^3 + 6(j\omega)^2 + 5(j\omega) + 30 = 0 \quad (2.28)$$

Factorizando la ecuación 2.28 se encuentra la ecuación 2.29.

$$6(5 - \omega^2) + j\omega(5 - \omega^2) = 0 \quad (2.29)$$

Despejando  $\omega$  de la ecuación 2.29 considerando que es necesario eliminar  $j$  por ser imaginario se obtiene  $\omega^2 = 5$  es decir  $\omega = \sqrt{5}$ , aplicando la ecuación 2.30 se encuentra el periodo de oscilación visto en la ecuación 2.31.

$$P_{cr} = \frac{2\Pi}{\omega} \quad (2.30)$$

$$P_{cr} = \frac{2\Pi}{\sqrt{5}} = 2,8099 \quad (2.31)$$

Considerando las reglas del cuadro 2.2 son encontradas las variables  $K_p$ ,  $Ti$  y  $Td$  mostradas a continuación.

$$K_p = 0,6 * K_{cr} = 0,6 * 30 = 18 \quad (2.32)$$

$$Ti = 0,5 * P_{cr} = 0,5 * 2,8099 = 1,405 \quad (2.33)$$

$$Td = 0,125 * P_{cr} = 0,125 * 2,8099 = 0,35124 \quad (2.34)$$

Reemplazando en la ecuación 2.17 los valores de  $K_p$ ,  $Ti$  y  $Td$  se halla la ecuación 2.35

$$PID = 18 \left( 1 + \frac{1}{1,405 * S} + 0,3124 * S \right) \quad (2.35)$$

Con el fin de obtener la forma convencional del controlador se simplifica la ecuación 2.35 se encuentra la ecuación 2.36.

$$PID = \frac{12,81 + 18 * S + 5,6232 * S^2}{S} \quad (2.36)$$

Simplificando la ecuación 2.36 es hallada la ecuación 2.37.

$$PID = \frac{6,3223 (S + 1,4235)^2}{S} \quad (2.37)$$

### 2.1.1.3. Lugar de las raíces

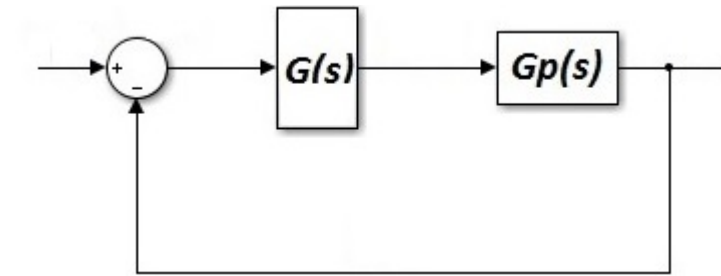
Para diseñar un sistema de control utilizando el lugar de las raíces según Katsuhiko Ogata [19] consiste en re-dibujar el lugar de las raíces agregando polos y ceros empleando la función de transferencia en lazo abierto tal que al dibujar el lugar de las raíces pase por los polos en el lazo cerrado del sistema deseados en el plano  $s$ . Esto se fundamenta en la hipótesis que el sistema posee en lazo cerrado un par de polos dominantes por lo que al agregar polos y ceros al sistema no afecta la respuesta.

El diseño adecuado del sistema de control empleando el lugar de las raíces requiere realizar un ajuste en la ganancia agregando un componente (polos y/o ceros) para reconstruir el lugar de las raíces a través de un compensador con el fin de localizar un par de polos dominantes en la posición deseada en lazo cerrado. Con el fin de encontrar el posicionamiento en el lugar de las raíces se debe considerar varios parámetros de diseño como es el sobre pico deseado, el tiempo de estabilización frente a un estímulo y el porcentaje de error.

Un ejemplo es el control de la planta  $Gp(s)$  expuesto en la ecuación 2.38 que es implementado en un sistema de lazo cerrado con retro-alimentación unitaria negativa como se muestra en la (*Figura 2.11*). Es requerido un tiempo de respuesta menor 10 segundos en lazo cerrado, un sobre pico menor al 10 % y un margen de error estacionario frente a un estímulo del sistema del tipo «step» menor al 5 %.

$$Gp(s) = \frac{4}{s^2 + 11,2 * s + 4} \quad (2.38)$$

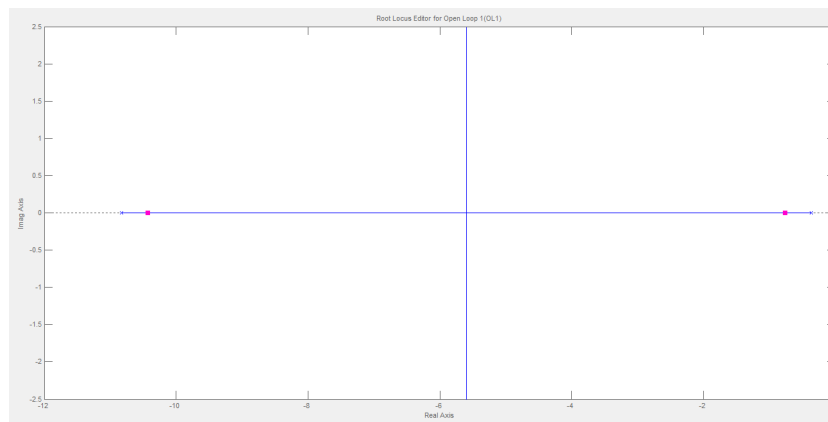
Figura 2.11: Sistema lazo cerrado



Fuente: Elaboración propia

Es localizado los polos y los ceros de la planta generando la (Figura 2.12) utilizando la herramienta que proporciona una interfaz gráfica para análisis y diseño «*rltool*» del programa *Matlab*®.

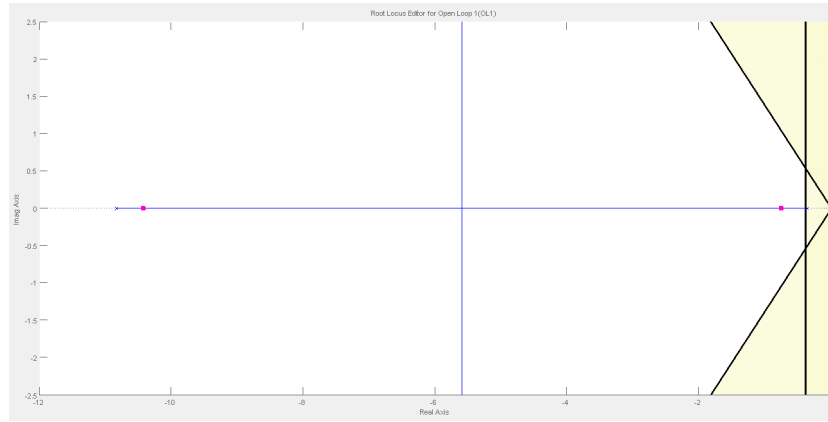
Figura 2.12: Polos y ceros de la planta



Fuente: Elaboración propia

Determinando una región de trabajo fijando los parámetros de diseño planteados anteriormente, se encuentra la (Figura 2.13)

Figura 2.13: Región de diseño



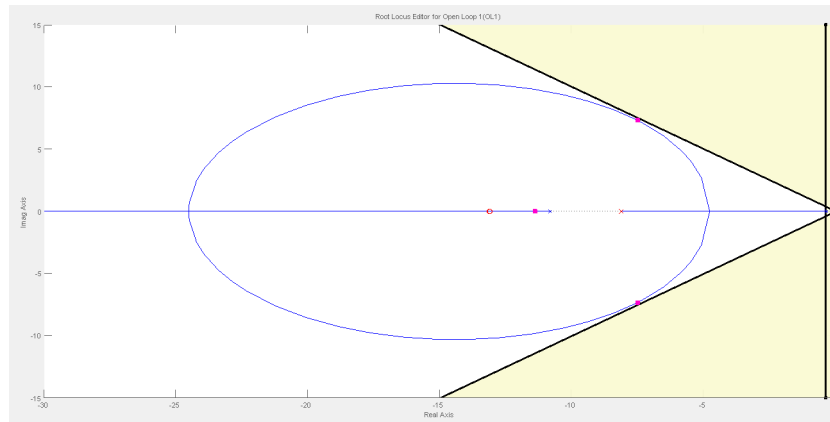
Fuente: Elaboración propia

Sabiendo que la forma del controlador PID en el lugar de las raíces (ecuación 2.39) el cual está compuesto por dos ceros, un polo y una constante de ganancia.

$$G(s) = \frac{A(s+B)(s+C)}{(s+D)} \quad (2.39)$$

Posicionando las componentes del controlador PID en la región de trabajo y ajustando los parámetros para que todos los componentes queden en la región permitida se encuentra la (Figura 2.14).

Figura 2.14: PID lugar de las raíces



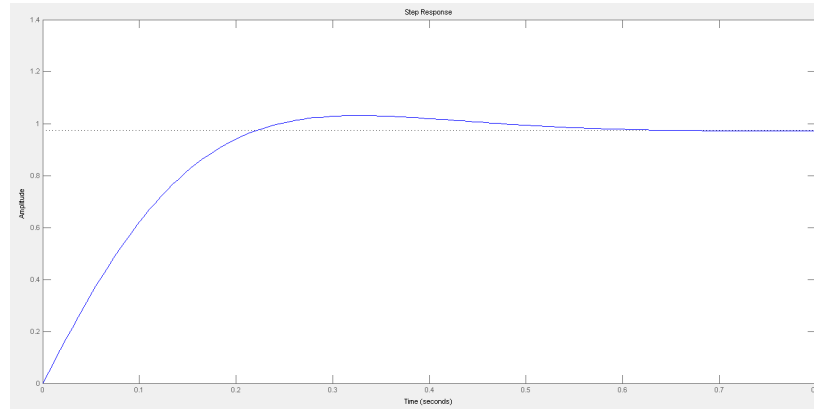
Fuente: Elaboración propia

Obteniendo la ecuación del PID mostrada en la ecuación 2.40 donde el valor de las variables resultantes es A igual 1.7813, B igual a 13.14, C igual 13.12 y D igual 8.144.

$$G(s) = \frac{1,7813 * (s + 13,14) (s + 13,12)}{(s + 8,144)} \quad (2.40)$$

La respuesta del sistema empleando el controlador encontrada anteriormente (ecuación 2.40) se obtiene la solución exhibida en la (Figura 2.15) donde tiene un tiempo de estabilización de 0.5 segundos y un sobre pico de 5,78 % cumpliendo con dos de los parámetros de diseño planteado anteriormente.

Figura 2.15: Respuesta del sistema frente a un estímulo del tipo step



Fuente: Elaboración propia

El error estacionario frente a un estímulo del tipo «step» se halla con base a la fórmula mostrada en la ecuación 2.41.

$$E_{ee} = \lim_{s \rightarrow 0} s * \left( \frac{1}{s} \right) \left( 1 - \frac{G(s) * G_p(s)}{1 + G(s) * G_p(s)} \right) \quad (2.41)$$

Reemplazando la ecuación 2.41 por los valores de la planta y el controlador encontrado es obtenida la ecuación 2.42. Comprobando que en el estado estacionario es igual a 2,581074596 % cumpliendo finalmente con todos los parámetros de diseños planteados para el ejemplo.

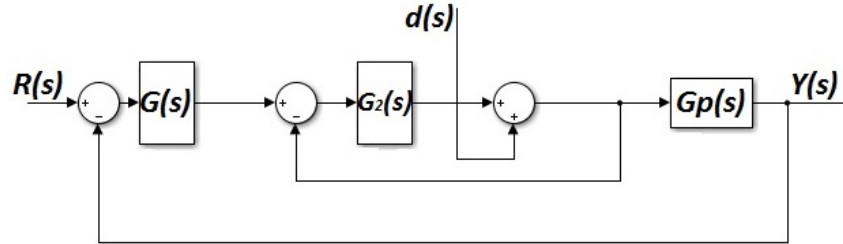
$$E_{ee} = \lim_{s \rightarrow 0} s * \left( \frac{1}{s} \right) \left( 1 - \frac{\frac{1,7813 * (s+13,14)(s+13,12)}{(s+8,144)} * \frac{4}{s^2+11,2*s+4}}{1 + \frac{1,7813 * (s+13,14)(s+13,12)}{(s+8,144)} * \frac{4}{s^2+11,2*s+4}} \right) = 0,02581074596 \quad (2.42)$$

### 2.1.2. Controlador PID en cascada

El controlador PID en cascada como es argumentado por Claudia Montero [4], es la configuración donde la salida de un controlador con retro-alimentación es la entrada de otro controlador con retro-alimentación que cubre todo el sistema como es visto en la (Figura 2.16), la cual contempla una perturbación denominada  $d(s)$ , y es asignado un controlador para la nueva entrada denominada  $G_2(s)$  con su respectiva retro-alimentación. Los parámetros  $G_2(s)$  y  $d(s)$  forman un sub-sistema dentro del sistema

principal compuesto por  $G(s)$  y  $G_p(s)$  por lo que la salida del controlador denominado  $G(s)$  es la entrada de sub-sistema ya mencionado; la salida de  $d(s)$  se convierte en la salida del sub-sistema y por ende la entrada de la planta  $G_p(s)$ .

Figura 2.16: Diagrama de bloques PID cascada



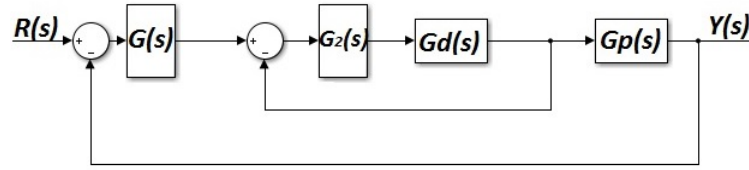
Fuente: Elaboración propia

Como es expuesto en la (Figura 2.16) el controlador en cascada es realizado con el fin de eliminar algunos efectos de las perturbaciones por medio de un segundo controlador para poder mejorar la dinámica del sistema. Para eliminar los efectos de la perturbación del sistema es necesario considerar las segundas variables creadas por el sub-sistema considerando las siguientes reglas:

- Diseñar el controlador del sub-sistema tal que contenga las perturbaciones más serias.
- Hacer el sub-sistema de respuesta rápida para disminuir los retrasos del sistema.
- Seleccionar las variables secundarias cuyos valores se encuentren relacionados a los valores de las variables primarias.
- Incluir en el sub-sistema la mayor cantidad de perturbaciones posibles presentadas en el sistema buscando la mayor velocidad de respuesta posible.
- Seleccionar una variable secundaria de control que permita al controlador del sub-sistema operar con la ganancia más alta.

Utilizando el concepto dado de PID en cascada dado en la (Figura 2.16) se plantea un sistema donde la perturbación  $d(s)$  es descrita como una función de transferencia por lo que se puede re escribir el diagrama de bloques reemplazando  $d(s)$  por  $Gd(s)$  obteniendo como resultado el diagrama de bloques expuesto en la (Figura 2.17).

Figura 2.17: PID cascada



Fuente: Elaboración propia

Para comprender de mejor manera el diseño en cascada es planteado un ejemplo en torno al diagrama encontrado en la (Figura 2.17) donde se desea controlar la planta encontrada en la ecuación 2.43, y sabiendo que la función que describe la perturbación del sistema es dada por la ecuación 2.44. Se desea que la respuesta del sistema sea igual a  $2s$  y se considera un zita igual a  $0.7$

$$Gp(s) = \frac{10}{s^2 + 7s + 10} \quad (2.43)$$

$$Gd(s) = \frac{4}{s^2 + 5s + 4} \quad (2.44)$$

Empleando la forma del PID exhibido en la ecuación 2.45 y sabiendo que la forma que adquiere la función de transferencia en lazo cerrado está dado por la ecuación 2.46 se encuentra que el nominador del sistema llamado en este caso  $B(s)$  que es igual a  $s^3 + 7s^2 + 10s + kp * kd * s^2 + kp * s + kp * ki$  que al agrupar en términos semejantes se obtiene la ecuación 2.47.

$$PID = \frac{kp * kd * s^2 + kp * s + kp * ki}{s} \quad (2.45)$$

$$Y(s) = \frac{Gp(s)}{1 + G(s) * Gp(s)} \quad (2.46)$$

$$B(s) = s + s^2(7 + kp * kd) + s(10 + kp) + kp * ki \quad (2.47)$$

Para encontrar las constantes del controlador con base a la ecuación característica del sistema, es necesario subir de grado multiplicándolo por el polinomio no dominante planteando la ecuación 2.48 que al agrupar términos es obteniendo la ecuación 2.49.

$$P(s) = (s^2 + 2 * \xi * Wn * s + (Wn)^2) * (s + 10 * \xi * (Wn)^2) \quad (2.48)$$

$$P(s) = s^3 + (12 * \xi * Wn * s)^2 + S((Wn)^2 + 20 * \xi^2 * (Wn)^2) + 10 * \xi * (Wn)^3 \quad (2.49)$$

Al igualar las ecuaciones 2.47 y 2.49 se obtienen las ecuaciones de los componentes del controlador PID en las ecuaciones 2.50, 2.51 y 2.52.

$$12 * \xi * Wn = 7 + kp * kd \quad (2.50)$$

$$(Wn)^2 + 20 * \xi^2 * (Wn)^2 = 10 + kp \quad (2.51)$$

$$10 * \xi * (Wn)^3 = kp * ki \quad (2.52)$$

También se sabe que la ecuación de la frecuencia natural es dada por la ecuación 2.53 donde T es el tiempo de respuesta deseado:

$$Wn = \frac{4,6}{\xi * T} \quad (2.53)$$

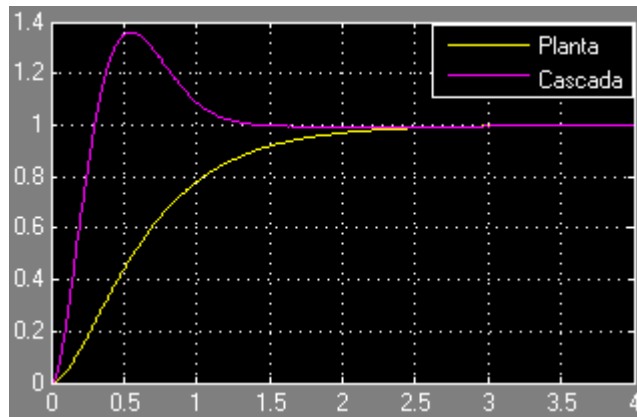
Empleando las ecuaciones 2.50, 2.51, 2.52 y 2.53 anteriormente vistas se obtiene el controlador  $G(s)$  mostrado en la ecuación 2.54. Aplicando los mismos conceptos pero fundamentado en la planta  $Gd(s)$  y aplicando las mismas condiciones planteadas inicialmente, considerando que en estado crítico la perturbación se estabiliza al mismo tiempo que la planta. Dando como resultado la ecuación  $G_2(s)$  expuesta en la ecuación 2.55.

$$G(s) = \frac{10,6000 * s^2 + 106,5959 * s + 248,3061}{s} \quad (2.54)$$

$$G_2(s) = \frac{13,3092 * s^2 + 112,5959 * s + 262,2826}{s} \quad (2.55)$$

Al reemplazar los controladores encontrados en las ecuaciones 2.54 y 2.55 en el diagrama de bloque de la (Figura 2.17) se obtiene el resultado mostrado en la (Figura 2.18) donde la respuesta en amarillo es el resultado en lazo abierto del sistema y el de color morado es la respuesta del sistema en lazo cerrado contemplando los controladores PID.

Figura 2.18: Respuesta frente PID en cascada



Fuente: Elaboración propia



### 2.1.3. Controlador PID discreto

Los controladores discretos surgen a partir de los controladores continuos vistos anteriormente con el fin de implementarse de forma digital como en los micro-controladores. En el trabajo de Steeve Erasmo Toledo Chojolán [3] titulado «DISEÑO DE CONTROLADORES PID EN TIEMPO DISCRETO, Y ANÁLISIS DE RESPUESTA UTILIZANDO HERRAMIENTAS COMPUTACIONALES» expone la necesidad de tener algunos conceptos claros para poder realizar análisis de los controladores discretos como el tiempo de muestreo y discretización que sería aplicada al controlador.

- tiempo de muestreo

Es el tiempo que muestreo es tiempo del sistema en el que demora en tomar una muestra y generar una respuesta. La exactitud de la respuesta del sistema discreto depende de la exactitud de la selección del tiempo, por lo tanto a menor el tiempo de muestre, mayor exactitud adquirirá el sistema pero requerirá mayor velocidad en el procesamiento por lo que es recomendado emplear 30 veces la frecuencia natural del sistema.

- discretización de controlador

Es el proceso matemático por el cual se pasa un sistema en tiempo continuo a tiempo discreto, el cual considera un tiempo de muestreo para evaluar la reacción del sistema, para lograr esto existen tres métodos para discretizar los controladores PID los cuales son: aproximación rectangular hacia delante (Euler I), aproximación rectangular hacia atrás (Euler II) y aproximación bilineal (trapezoidal o Tustin).

En el cuadro 2.3 se observan algunas ventajas y desventajas de los métodos de discretización del PID en el que se destaca el método de aproximación bilineal (trapezoidal o Tustin) el cual presenta mayor precisión y estabilidad frente al control del sistema.

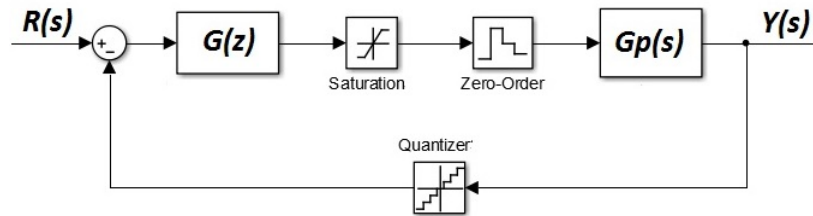
Cuadro 2.3: Comparación controladores PID discreto

Método de discretización PID	Ventajas	Desventajas
Rectangular hacia delante (Euler I)	Genera respuesta más rápida del sistema	En tiempo de muestreo inadecuado el sistema tiende a ser inestable.
Aproximación rectangular hacia atrás (Euler II)	Genera una respuesta más segura frente a la respuesta del sistema	Genera atrasos en la respuesta del sistema
Aproximación bilineal (trapezoidal o Tustin)	Es exacto frente a la respuesta del sistema	Requiere mayor capacidad de procesamiento

Fuente: Elaboración propia

Un ejemplo de aplicación PID discreto mostrado a continuación aplicado en un sistema expuesto en la (Figura 2.19), utilizando la herramienta gráfica de *Matlab®/Simulink*. Donde  $R(s)$  corresponde a la entrada de sistema,  $G(z)$  es el controlador PID digital utilizando el método de aproximación bilineal aplicado en el controlador  $G(s)$ , utiliza un diagrama de bloques de saturación del sistema donde es fijada la respuesta máxima y mínima permitida en este caso, el bloque de ordenador de orden cero o (*Zero-Order*) para poder controlar la planta  $Gp(s)$  en tiempo continuo y obtener la salida en tiempo continuo  $Y(s)$  y por último utiliza una re-alimentación utilizando el bloque (*Quantizer*) donde se indica el valor de la entrada que correspondería a 256 equivalente a un «bite».

Figura 2.19: Sistema lazo cerrado con PID digital



Fuente: Elaboración propia

La planta  $Gp(s)$  planteada para este ejercicio es la expuesta en la ecuación 2.56 y el PID análogo aplicado en este caso visto en la ecuación 2.57.

$$Gp(s) = \frac{10}{s^2 + 7 * s + 10} \quad (2.56)$$

$$G(s) = 106,5959 * \left( 1 + \frac{2,3294}{s} + 0,1933 * s \right) \quad (2.57)$$

Empleando la forma más común del controlador PID mostrada en la ecuación 2.58 se determina la transformación transitoria expuesta en la ecuación 2.59 que es usada para obtener la respuesta en caso discreto que adquiere la forma reflejada en la ecuación 2.60.

$$U(t) = Kp * \left( E(t) + \frac{1}{Ti} \int_0^t E(t) dt + Td \frac{dE(t)}{dt} \right) \quad (2.58)$$

La respuesta transitoria en ecuaciones diferenciales de la ecuación 2.58 es determinada en la ecuación 2.59.

$$Uk = U(k-1) + q0 * Ek + q1 * E(k-1) + q2 * E(k-2) \quad (2.59)$$

Aplicando la transformada Z a la ecuación 2.59 es obtenida la ecuación 2.60.

$$\frac{U(z)}{E(z)} = \frac{q0 + q1 * z^{-1} + q2 * z^{-2}}{1 - z^{-1}} \quad (2.60)$$

Al haber seleccionado el método de aproximación bilineal se aplica la ecuaciones 2.61, 2.62 y 2.63 que corresponden a los parámetros de las variables expuestas en el PID discreto mostrado en la ecuación 2.60.

$$q0 = Kp * \left( 1 + \frac{T}{2 * Ti} + \frac{Td}{T} \right) \quad (2.61)$$

$$q1 = Kp * \left( \frac{T}{2 * Ti} - 1 - 2 * \frac{Td}{T} \right) \quad (2.62)$$

$$q2 = Kp * \frac{Td}{T} \quad (2.63)$$

Conociendo la frecuencia natural del sistema es de 3.2857 Hz considerando que el sistema emplea un zita de 0.7 y se desea un tiempo de estabilización de 2 segundos; que el tiempo de muestreo recomendado es de 30 veces la frecuencia natural se encuentra que el tiempo de muestreo es de 0.0266 segundos como se expresa en la ecuación 2.64 donde T es el tiempo de muestreo en segundos.

$$T = \frac{1}{3,2857 * 30} = 0,0101 \quad (2.64)$$

Por definición del PID mostrado en la ecuación 2.58 y al ser comparado con el PID visto en la ecuación 2.57 se obtiene que por definición la variable  $Ti$  es igual  $\frac{1}{Ki}$  y la variable  $Td$  es igual a  $Kd$  como se demuestra en las ecuaciones 2.65 y 2.66.

$$Ki = \frac{1}{Ti} \implies Ti = \frac{1}{Ki} \quad (2.65)$$

$$Kd = Td \quad (2.66)$$

Por lo tanto al reemplazar por los valores del controlador PID continuo, visto en la ecuación 2.57 da como resultado las ecuaciones.

$$Ti = \frac{1}{2,3294} = 0,4293 \quad (2.67)$$

$$Td = 0,1933 \quad (2.68)$$

Ya teniendo las incógnitas para encontrar el PID discreto por el método de aproximación bilineal son remplazadas las variables en las ecuaciones 2.61, 2.62 y 2.63 considerando el valor de  $Kp = 106,5959$  obteniendo las ecuaciones 2.69, 2.70 y 2.71.

$$q0 = 106,5959 * \left( 1 + \frac{0,0101}{2 * 0,4293} + \frac{0,1933}{0,0101} \right) = 2138,4 \quad (2.69)$$

$$q1 = 106,5959 * \left( \frac{0,0101}{2 * 0,4293} - 1 - 2 * \frac{0,5078}{0,0101} \right) = -4166,5 \quad (2.70)$$

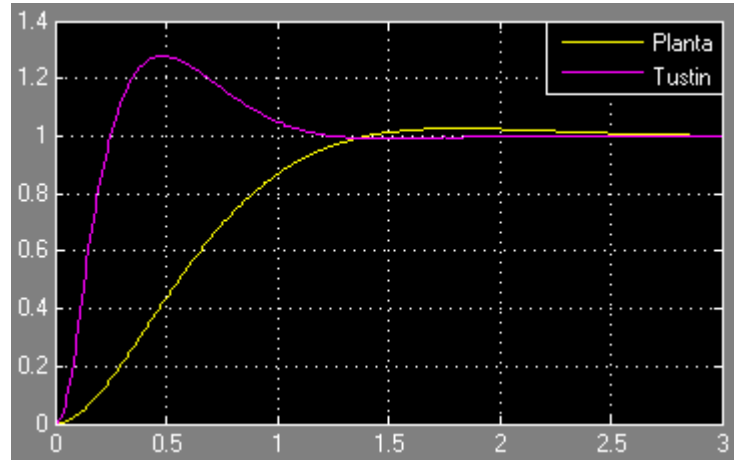
$$q2 = 106,5959 * \frac{0,1933}{0,0101} = 2030,6 \quad (2.71)$$

Al reemplazar en la ecuación 2.60 es obtenida la ecuación 2.72 correspondiente al PID en discreto utilizado para el control de la planta es decir  $G(z)$ .

$$G(z) = \frac{2138,4 - 4166,5 * z^{-1} + 2030,6 * z^{-2}}{1 - z^{-1}} \quad (2.72)$$

Obteniendo la respuesta en discreto exhibida en la (*Figura 2.20*) en la que se puede ver la respuesta del sistema en lazo cerrado aplicando el controlador PID en discreto por el método de de aproximación tustin de color amarillo y la respuesta natural de la planta en lazo abierto de color morado.

Figura 2.20: Respuesta del sistema con el controlador PID discreto



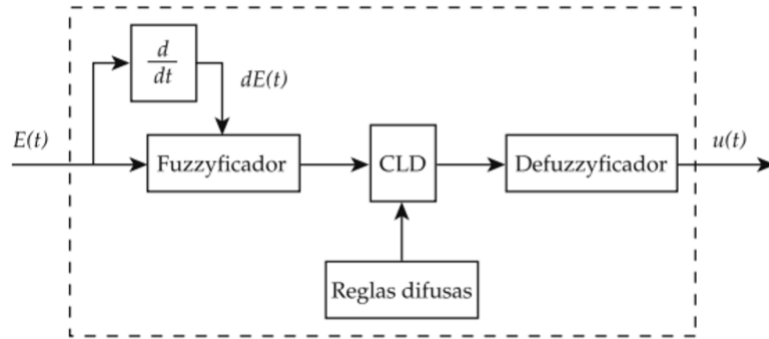
Fuente: Elaboración propia

#### 2.1.4. Controlador difuso

El Controlador difuso o también llamado controlador «*Fuzzy*» por su nombre en inglés, es fundamentado en la lógica difusa, al ser una herramienta utilizada para la toma de decisiones utilizando variables del tipo análogo en el espacio continuo y pasándolo a valores digitales en el espacio discreto, empleados en sistemas electrónicos como las computadores que trabajan con base a valores fijos. También, en el trabajo de Natalie Segura Velandia y Oswaldo Rivera Rincon [30] es argumentado que la lógica difusa nace de la necesidad de expresar a través de un lenguaje matemático algunas situaciones de la vida cotidiana ocasionado la necesidad de realizar un análisis o desarrollo. Debido a esto se entiende que el objetivo del control difuso es crear un tipo de inteligencia artificial entorno a múltiples variables y resultados obtenidos en pruebas anteriores.

La arquitectura de los controladores difusos son iguales entre si independientemente de la planta o proceso que se desea controlar, al cambiar solo las reglas del controlador difuso, por lo que se puede plantear un controlador difuso genérico exhibido en la (Figura 2.21) exponiendo la entrada del controlador la cual es denotada con  $E(t)$ , que es el error de la respuesta y la variación de la respuesta debido que con esos datos se determina el comportamiento del sistema.

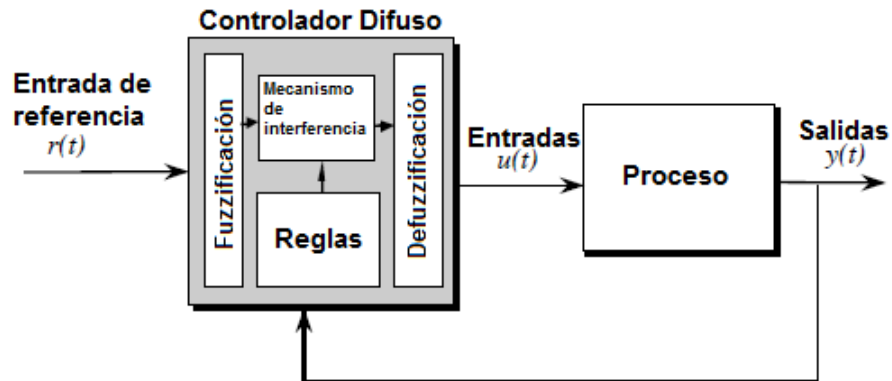
Figura 2.21: Controlador fuzzy genérico



Fuente: Natalie Segura Velandia y Oswaldo Rivera Rinco[30]

En la (Figura 2.22) se visualiza un controlador difuso genérico a través de un diagrama de bloques extraído del libro de Kevin M y Yurkovich [20] «*Control Fuzzy*» donde se exhibe un controlador difuso cuya entrada de referencia está denotada como  $r(t)$  y su salida es denotada por  $u(t)$ ; la salida del controlador es conectada como la entrada de la planta (*Process*) y la salida es denotada por  $y(t)$  finalizando con una retro-alimentación al controlador formando un sistema en lazo cerrado.

Figura 2.22: Arquitectura controlador fuzzy



Fuente: Kevin M y Yurkovich[20]

Para realizar el controlador genérico mostrado en las (Figuras 2.21 y 2.22) es necesario establecer cuatro parámetros explicados en [30, 20] los cuales son:

**Reglas difusas:** Son las reglas que permiten controlar el sistema siendo originadas en las «*Operaciones borrosas*» (*and, or, not*) que realiza una evaluación de las reglas teniendo en cuenta los antecedentes usando operaciones lógicas en controladores y modelos difusos. Después se realizan «*Operaciones de membresía*» en que se define el grado de pertenencia de un elemento a un subconjunto. Las reglas difusas en un

sistema se determinan de acuerdo a la salida que se desee obtener, estas reglas se ajustan con respecto a las entradas del sistema, ajustándolas de tal manera que la salida que se va a obtener sea la deseada. Un claro ejemplo se observa en las figuras (*Figura 2.23* y 2.24), donde se muestran las reglas para el comportamiento de la velocidad del motor para el controlador difuso, se observan las dos entradas (Velomoto y error) y la respectiva salida deseada en cada una de las reglas.

Figura 2.23: Reglas para el comportamiento de la velocidad de un motor

1. If (velomoto is min) and (error is negativo) then (salida is velomedbaja) (1)
2. If (velomoto is regular) and (error is negativo) then (salida is velomedbaja) (1)
3. If (velomoto is max) and (error is negativo) then (salida is velaalta) (1)
4. If (velomoto is media) and (error is negativo) then (salida is velobaja) (1)
5. If (velomoto is alta) and (error is negativo) then (salida is velomediala) (1)
6. If (velomoto is min) and (error is aroximado) then (salida is velomedbaja) (1)
7. If (velomoto is regular) and (error is aroximado) then (salida is velomediala) (1)
8. If (velomoto is max) and (error is aroximado) then (salida is velomedalta) (1)
9. If (velomoto is media) and (error is aroximado) then (salida is velaalta) (1)
10. If (velomoto is alta) and (error is aroximado) then (salida is velomedalta) (1)
11. If (velomoto is min) and (error is positivo) then (salida is velobaja) (1)
12. If (velomoto is regular) and (error is positivo) then (salida is velomediala) (1)
13. If (velomoto is max) and (error is positivo) then (salida is velomedalta) (1)
14. If (velomoto is media) and (error is positivo) then (salida is velaalta) (1)
15. If (velomoto is alta) and (error is positivo) then (salida is velomedalta) (1)

Fuente: Elaboración propia.

Figura 2.24: Reglas para el comportamiento de la velocidad del motor

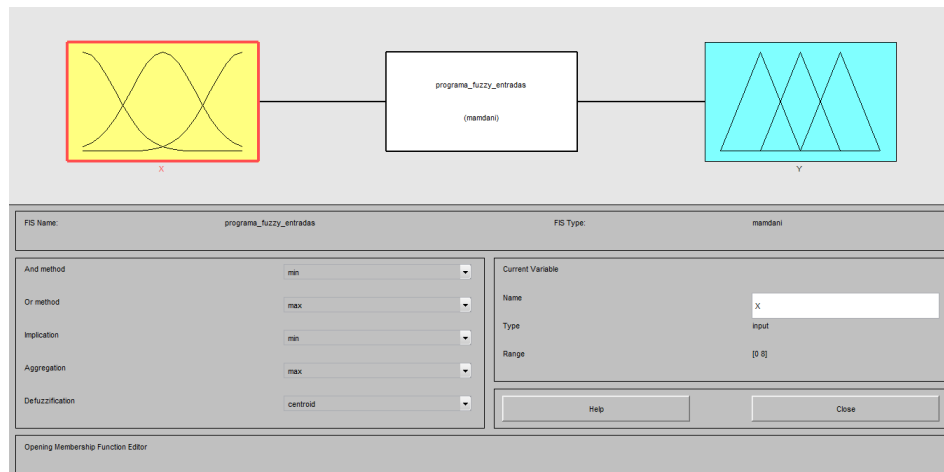


Fuente: Elaboración propia

**CLD o interfaz de mecanismo:** Evalúa cual de las reglas difusas es la adecuada para ser aplicada en el momento y entonces decide cual podría ser la salida de la planta.

**Fusificación:** Es una operación que sucede en todo instante de tiempo. Busca modificar la entrada del controlador de tal forma que puedan asignar los valores en las funciones de membresía. Un ejemplo de fusificación se observa en un sistema con una entrada y una salida como se observa a continuación en la (Figura 2.25) utilizando *Matlab®/Fuzzy*.

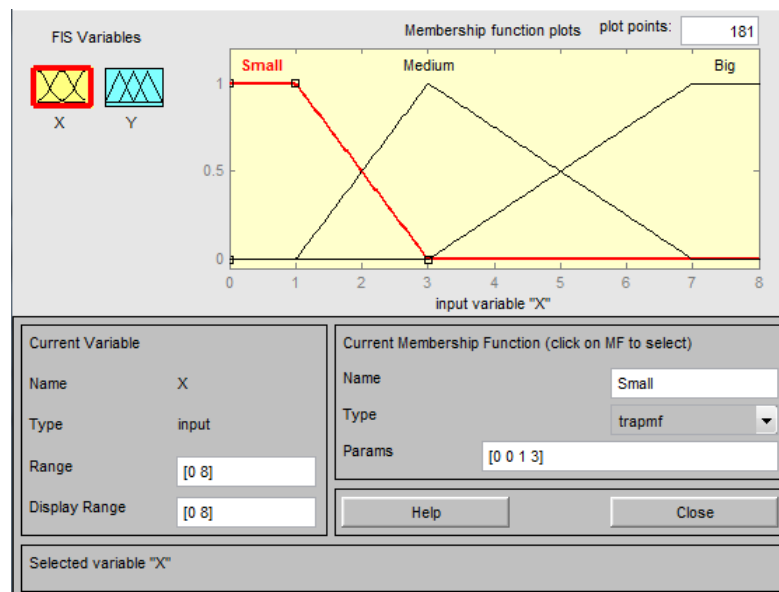
Figura 2.25: Sistema de ejemplo



Fuente: Elaboración propia

Además, se tienen las siguientes funciones de membresía

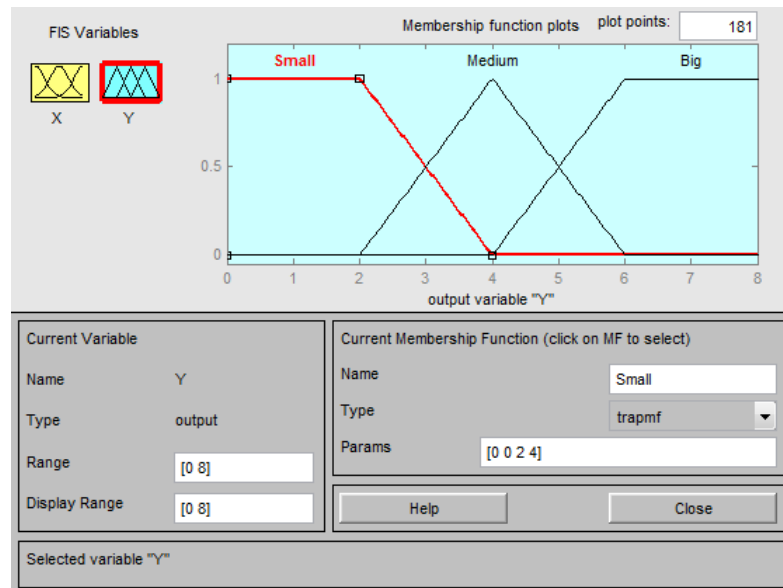
Figura 2.26: Función de membresía x



Fuente: Elaboración propia



Figura 2.27: Función de membresía en y



Fuente: Elaboración propia

Teniendo en cuenta las siguientes reglas

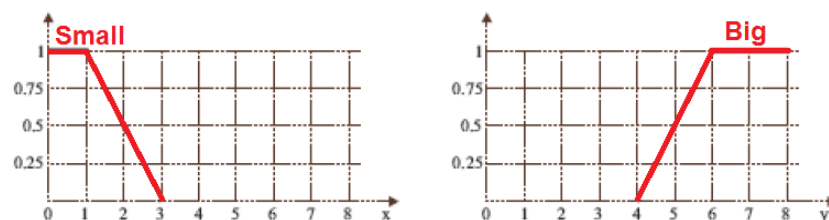
Figura 2.28: Reglas del sistema

1. If (X is Small) then (Y is Big) (1)
2. If (X is Medium) then (Y is Small) (1)
3. If (X is Big) then (Y is Medium) (1)

Fuente: Elaboración propia

- Si x es «*Small*» entonces y es «*Big*»

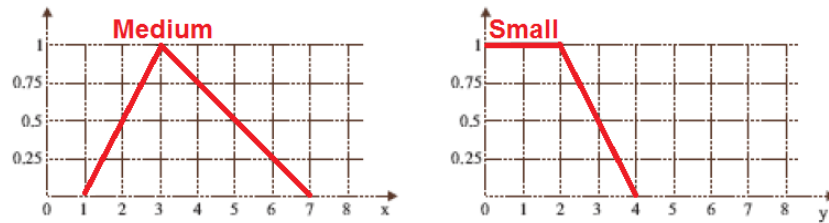
Figura 2.29: Regla #1



Fuente: Elaboración propia.

- Si x es «*Medium*» entonces y es «*Small*»

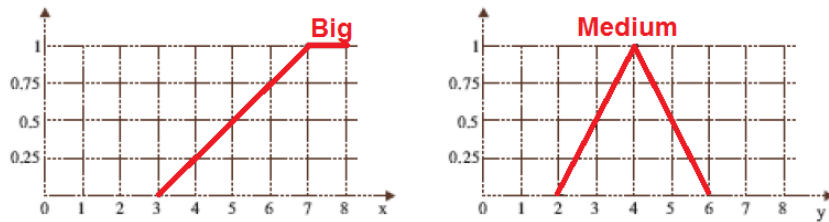
Figura 2.30: Regla #2



Fuente: Elaboración propia.

- Si  $x$  es «*Big*» entonces  $y$  es «*Medium*»

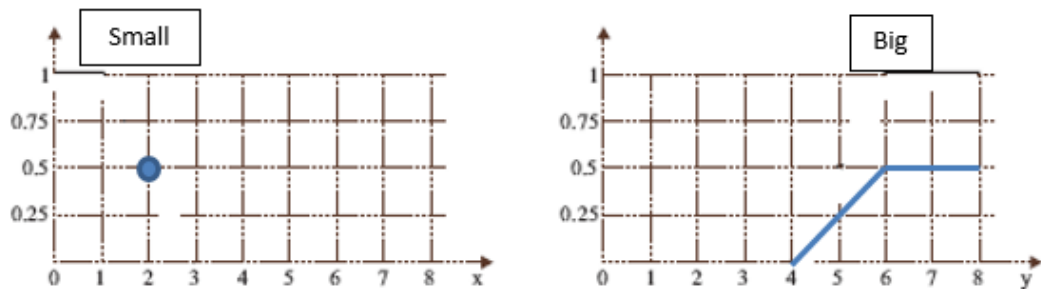
Figura 2.31: Regla #3



Fuente: Elaboración propia.

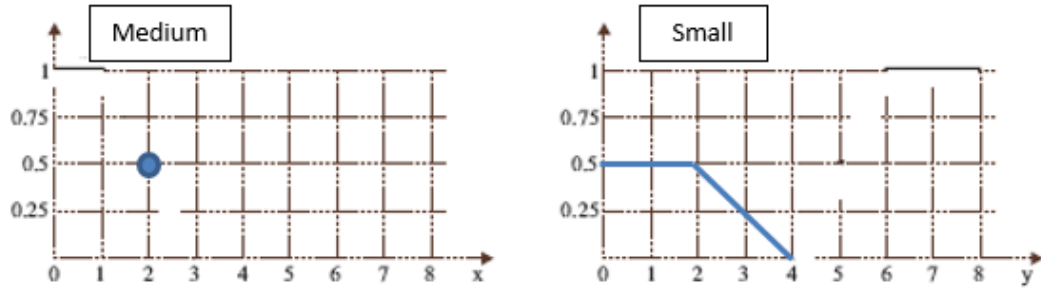
Realizando el proceso de fusificación, se procede a tomar un valor para la entrada  $x$ , que en este caso será  $x=2$ . Posteriormente, se continúa evaluando cada una de las anteriores reglas mencionadas en la entrada  $x=2$  y así se halla la respectiva salida para cada una de las reglas. Este proceso es realizado como se observa a continuación.

- Para la primera regla, se evalúa en  $x=2$  y se halla la respectiva salida, como se observa en la (Figura 2.32) vista a continuación.

Figura 2.32: Regla # 1 evaluada en  $x=2$ 

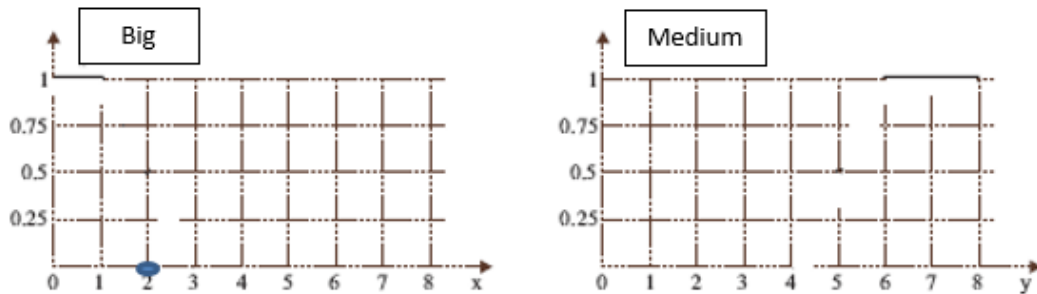
Fuente: Elaboración propia

- Para la segunda regla, se evalúa en  $x=2$  y se halla la respectiva salida, como se observa en la (Figura 2.33) vista a continuación.

Figura 2.33: Regla # 2 evaluada en  $x=2$ 

Fuente: Elaboración propia

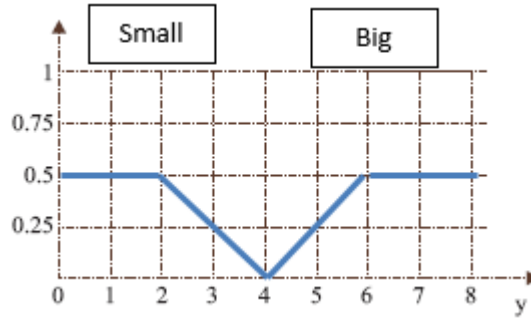
- Para la tercera regla, se evalúa en  $x=2$  y se halla la respectiva salida, como se observa en la (Figura 2.34) vista a continuación.

Figura 2.34: Regla # 3 evaluada en  $x=2$ 

Fuente: Elaboración propia

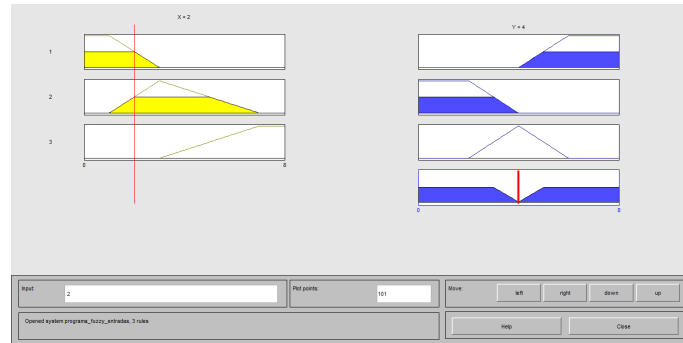
Luego, ya habiendo obtenido las respectivas salidas en cada una de las reglas, se unen cada una de las gráficas de salida en  $y$  para obtener la gráfica o conjunto resultante. Este conjunto resultante se puede ver en la (Figura 2.35).

Figura 2.35: Conjunto resultante



Fuente: Elaboración propia

Finalmente, se compara el conjunto resultante obtenido de forma teórica con el conjunto resultante que se obtiene del programa en «*Fuzzy*» visto en la (Figura 2.36) vista a continuación.

Figura 2.36: Conjunto resultante en «*Fuzzy*»

Fuente: Elaboración propia

Por último, al realizar la comparación de las gráficas o conjuntos resultantes se observa que las gráficas son las mismas y que finalmente se obtiene una salida en  $y=4$

**Defusificación:** Es el proceso matemático en que convierte la conclusión difusa obtenida a partir de la entrada en números reales que se vuelven la salida del controlador y por ende la entrada de la planta ( $u(t)$ ). Según [14] existen diferentes métodos de defusificación, los cuales son:

- Método de criterio máximo: Este método busca que el resultado se ubica en el punto cuando la distribución de acciones de control llega a su máximo punto.
- Método del promedio máximo: Este método establece que se crea una acción de control que está dada por la media o promedio de todas las acciones de control cuando los grados de pertenencia alcanzan el valor máximo. La salida puede definirse por la ecuación 2.73 que se observa a continuación.

$$Z_0 = \sum_{j=1}^I \frac{W_j}{I} \quad (2.73)$$

En esta fórmula  $I$  es el número de etiquetas que alcanzan el valor máximo y la función de membresía alcanza un valor máximo de  $U_z(W_j)$ .

- Método de centro de áreas: Este método se hace con el fin de encontrar una señal de control, la cual se halla calculando el centro de gravedad de la distribución de las salidas. En el caso discreto, las salidas se obtienen a través de la ecuación 2.74 que se muestra a continuación.

$$Z_0 = \frac{\sum_{j=1}^n U_{zz}(W_j) * W_j}{\sum_{j=1}^n U_z(W_j)} \quad (2.74)$$

Donde  $U_z$  es una función de pertenencia resultante,  $n$  es el número de niveles de cuantización.

Tomando el método de centro de áreas y continuando el ejemplo visto anteriormente en la parte de fusificación, con una entrada  $x = 2$  utilizando la (*Figura 2.35*), se calcula la salida defusificada en el dominio discreto de salida  $Y = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$  con la fórmula mencionada en la ecuación 2.74 tomando los puntos de intersección en cada uno de los puntos como es mostrado en la ecuación 2.75.

$$Salida = \frac{(0 * 0,5) + (1 * 0,5) + (2 * 0,5) + (3 * 0,25) + (4 * 0) + (5 * 0,25) + (6 * 0,5) + (7 * 0,5) + (8 * 0,5)}{0,5 + 0,5 + 0,5 + 0,25 + 0 + 0,25 + 0,5 + 0,5 + 0,5} = 4 \quad (2.75)$$

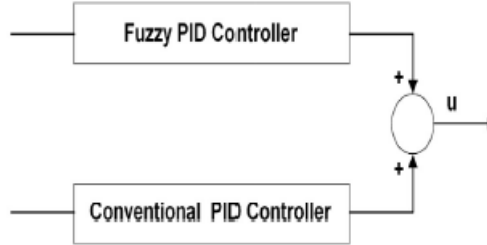
De esta forma se confirma que la salida es igual a cuatro como también se observa en la (*Figura 2.36*).

### 2.1.5. Controlador PID-Fuzzy

El controlador PID-Fuzzy es un nuevo tipo de controlador híbrido inteligente entendido entre los controladores PID y difuso con el fin de combinar las cualidades de cada controlador. Como es expresado por Erenoglu I.et.al. [9] en su trabajo titulado «*AN INTELLIGENT HYBRID FUZZY PID CONTROLLER*», el controlador PID presenta un control adecuado para sistemas lineales y es normalmente seleccionado al evaluar factores de costos/beneficio, pero presenta limitaciones frente a sistemas no lineales, mientras que el control difuso es utilizado exitosamente en sistemas no lineales gracias a sus características basadas en el conocimiento. Observando estos dos argumentos de los controladores es posible crear un nuevo controlador híbrido que compensa las falencias de cada uno.

Para exponer las cualidades de respuesta del controlador híbrido se combinan las señales como es expuesto en la (Figura 2.37) con el fin de analizar la respuesta del sistema frente a estímulo.

Figura 2.37: Diagrama de bloques PID-fuzzy



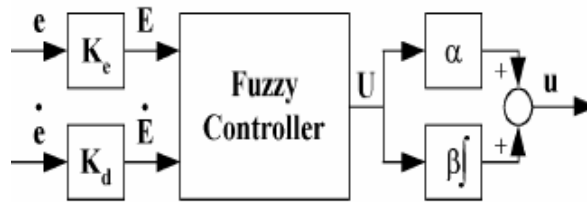
Fuente: Erenoglu I.et.al.[9]

Con el fin de comparar las respuestas de los controladores PID y PID-Fuzzy se implementa el controlador PID en su forma ideal vista en la ecuación 2.76, donde  $K_p$  es la constante proporcional,  $T_i$  es el tiempo de integración del sistema y  $T_d$  es el tiempo de derivación.

$$PID = K_p * \left( 1 + \frac{1}{T_i * s} + T_d s \right) \quad (2.76)$$

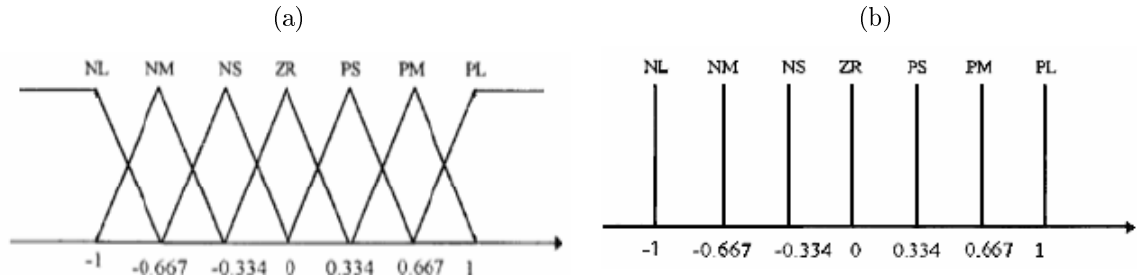
La estructura utilizada para la implementación del controlador PID-fuzzy es la expuesta en la (Figura 2.38) visualizando las dos entradas a sistema, los cuales son el error ( $e$ ) y el cambio del error ( $\dot{e}$ ), también se presenta en la (Figura 2.39) donde es mostrada la función de pertenencia de la entrada (Figura 2.39a) y la función de pertenencia de la salida (Figura 2.39b).

Figura 2.38: Controlador Fuzzy



Fuente: Erenoglu I.et.al. [9]

Figura 2.39: Funciones de pertenencia



Fuente: Erenoglu I.et.al. [9], 2.39a Función de pertenencia de entrada, 2.39b Función de pertenencia de salida.

Al emplear el controlador PID-fuzzy y el PID se fijan los parámetros de diseño que en caso del controlador PID es  $K_p$ ,  $T_i$  y  $T_d$ ; y para el controlador híbrido PID-Fuzzy se plantean unas reglas expuestas en el cuadro 2.4, para la implementación fuzzy del sistema planteado junto con otros parámetros  $K_e$ ,  $K_d$ ,  $\alpha$  y  $\beta$ .

Cuadro 2.4: Reglas controlador PID-Fuzzy

E/E	NL	NM	NS	ZR	PS	PM	PL
PL	ZR	ps	pm	pl	pl	pl	pl
PM	ns	ZR	ps	pm	pl	pl	pl
PS	nm	ns	ZR	ps	pm	pl	pl
ZR	nl	nm	ns	ZR	ps	pm	pl
NS	nl	nl	nm	ns	ZR	ps	pm
NM	nl	nl	nl	nm	ns	ZR	ps
NL	nl	nl	nl	nl	nm	ns	ZR

Fuente: Erenoglu I.et.al.[9]

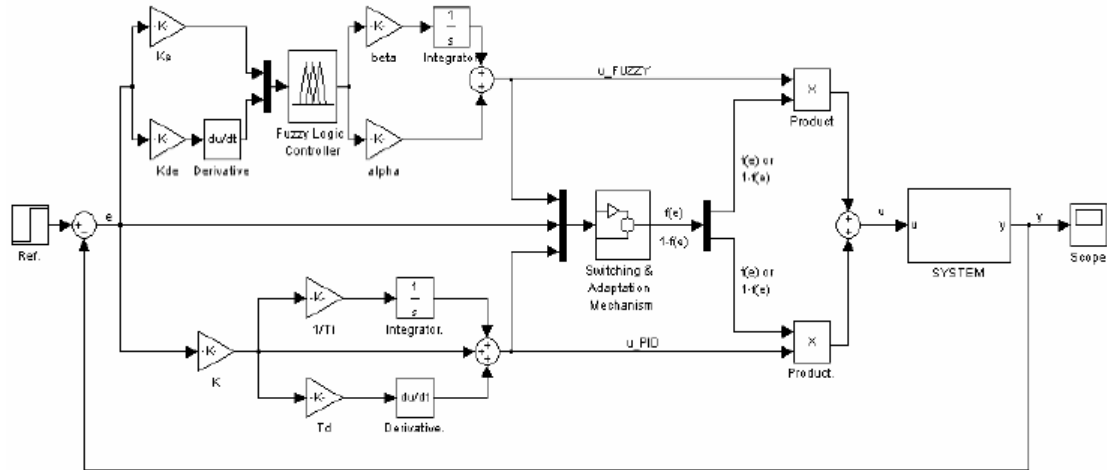
Por cuestiones de espacio fueron usadas abreviaciones para indicar el tipo de respuesta donde: ZR corresponde a cero o «Zero» en inglés, NL correspondiente a las palabras en inglés de «*Negative Longe*» que hacen referencia a la respuesta negativa grande, NM correspondiente a las siglas en inglés «*Negative medium*» haciendo referencia a la respuesta negativa mediana, NS correspondiente a «*Negative Small*» en inglés que hace referencia a la respuesta negativa pequeña, y de igual manera es aplicado a las respuestas positivos como PS correspondiente a «*Positive Small*» haciendo referencia a la respuesta positiva pequeña, PM correspondiente a «*Positive medium*» haciendo referencia de respuesta positiva mediana y finalmente PL correspondiente a «*Positive Longe*» que hace referencia a la respuesta positiva grande.

El controlador PID-Fuzzy y PID es implementado empleando diagrama de bloques usando *Matlab®/Simulink*, debido esto se implementa el bloque de «*switching & blending mechanism*» ya que determina el dominio de una de las dos estructuras, después

cualquiera de las dos respuestas es multiplicada por una de las funciones  $1 - f(e)$  y  $f(e)$  que permiten definir el nivel de actividad del controlador y así permite alcanzar un nivel razonable del equilibrio entre las acciones de los controladores.

Aplicando los controladores PID-Fuzzy y PID por el diagrama de bloques de *Matlab®* es encontrada la arquitectura expuesta en la (Figura 2.40).

Figura 2.40: Arquitectura controlador híbrido PID-Fuzzy



Fuente: Erenoglu I.et.al. [9]

Posteriormente, se realizaron dos simulaciones planteadas continuación:

- Primera simulación: Se realiza con un sistema de primer orden con un tiempo muerto teniendo en cuenta la ecuación 2.77.

$$G_1(s) = \frac{1}{s+1} e^{-0,2s} \quad (2.77)$$

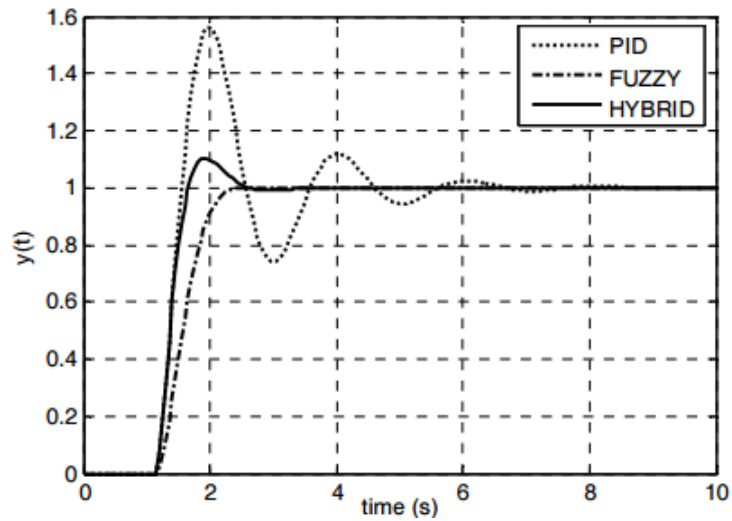
Los parámetros planteados para el PID son  $K=2$ ,  $T_i=0.25$ ,  $T_d=0.025$  y los parámetros del controlador PID-Fuzzy son  $\alpha = 0,05$ ,  $\beta = 4,5$ ,  $K_e = 1$ ,  $K_d = 0,56$ . Las salidas de los controladores y las respuestas del sistema se observan en la (Figura 2.41).

En la (Figura 2.41) se observan las diferentes respuestas de la planta frente a diferentes controladores con un estímulo del tipo «step»; se puede ver que la respuesta del controlador PID presenta un sobrepico mayor a comparación de los otros controladores, pero con un tiempo estabilización mas lento a comparación, mientras que el controlador híbrido tiene una respuesta donde hay un pequeño sobrepico mínimo junto con un tiempo de estabilización más rápido que el controlador PID. Por último, el controlador «fuzzy» posee un tiempo de estabilización más rápido entre los tres controladores vistos y no presenta ningún sobrepico en este caso.

Otra respuesta es la presentada en la (Figura 2.42) que visualiza las salidas de los controladores utilizados.

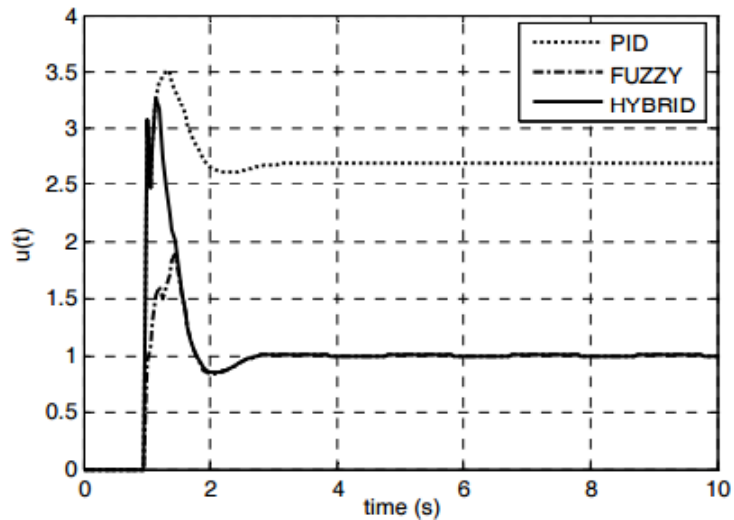


Figura 2.41: Primera simulación



Fuente: Erenoglu I.et.al.[9]

Figura 2.42: Salida de los controladores primera simulación



Fuente: Erenoglu I.et.al.[9]

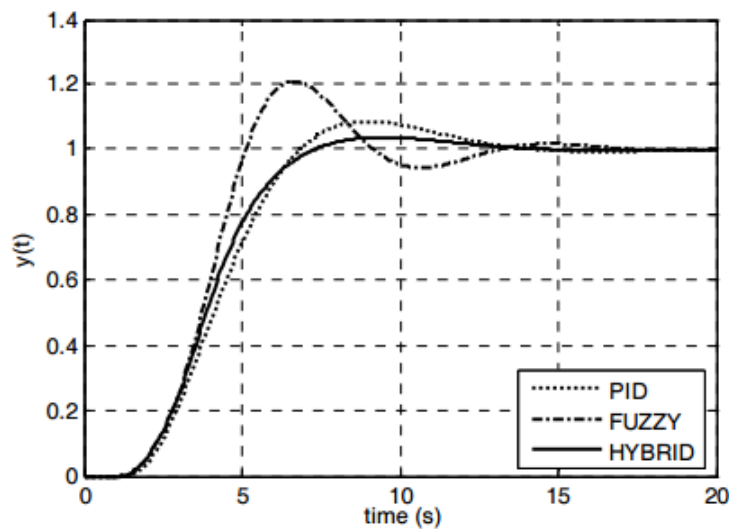
En la (Figura 2.42) anteriormente vista se puede ver que el controlador PID presenta mayor ganancia que los otros dos controladores; mientras tanto el controlador híbrido tiene un sobrepico más grande de todos los controladores pero tiene una menor ganancia con respecto al controlador PID, pero al estar compuesto por los dos componentes (PID y Fuzzy) tiene una mejor respuesta que los otros dos controladores y por último, el controlador «fuzzy» presenta un menor sobrepico y una ganancia igual al controlador híbrido, pero no tiene la respuesta esperada ante este estímulo.

- Segunda simulación: Esta simulación presenta un sistema de segundo orden y tiene función de transferencia vista en la ecuación 2.78.

$$G_2(s) = \frac{1}{(s+1)(s+2)} e^{-0,2s} \quad (2.78)$$

Los parámetros para el PID son  $K=0.01$ ,  $T_i=0.014$  y  $T_d=1$  y los parámetros para el controlador PID-Fuzzy son  $\alpha=0,2$ ,  $\beta=1,1$ ,  $K_e=1,5$  y  $K_d=0,2$ . Las salidas de los controladores y las respuestas del sistema se observan en la (Figura 2.43).

Figura 2.43: Segunda simulación

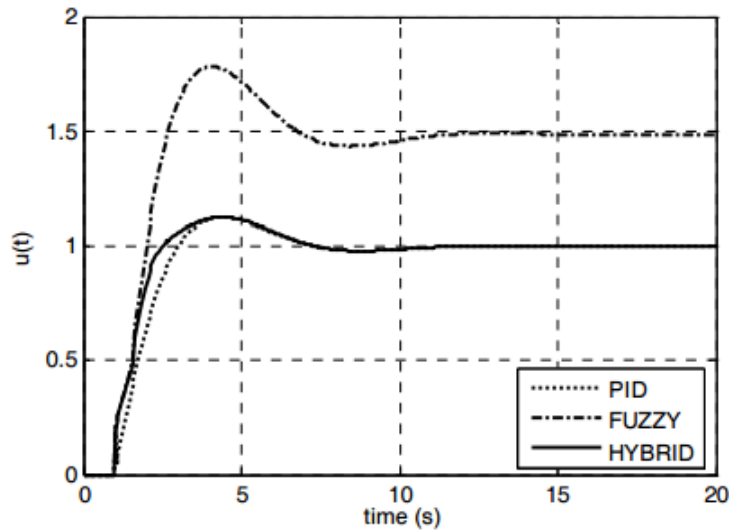


Fuente: Erenoglu I.et.al.[9]

En la (Figura 2.43) se expone que el controlador «fuzzy» tiene un mayor sobrepico a comparación de los demás controladores según al tiempo de estabilización más lento por lo que no tiene la respuesta deseada; el controlador híbrido posee un sobrepico mínimo y tiene un tiempo de estabilización rápido; por último, y en este caso el controlador que presenta la respuesta más óptima es el PID ya que presenta un pequeño sobrepico y un tiempo de estabilización rápido.

Además, en la (Figura 2.44) se pueden ver las salidas de cada uno de los controladores.

Figura 2.44: Salida de controladores segunda simulación



Fuente: Erenoglu I.et.al. [9]

En la (Figura 2.44) se tiene que el controlador «fuzzy» presenta una mayor ganancia y un mayor sobrepico que los demás controladores; los controladores híbrido y PID presentan una respuesta muy similar donde presentan prácticamente el mismo sobrepico y el mismo tiempo de estabilización, por lo que son los dos controladores de mejor respuesta entre los tres controladores comparados.

Los resultados de los casos presentados expone la constancia en la respuesta deseada por parte de controlador PID-Fuzzy ocasionando que independientemente del tipo de sistema planteado se obtenga un control adecuado a diferencia del controlador PID y difuso que presentan una respuesta frente al estímulo dependiendo del tipo de sistema implementado.

### 2.1.6. Comparación de controladores

Realizando un estudio comparativo entre todos los controladores, se realizó el siguiente cuadro

Cuadro 2.5: Comparación de controladores

CONTROLADOR	VENTAJAS	DESVENTAJAS
PID (Proporcional Integrativo Derivativo)	Es compatible con diferentes tipos de variables. Además, puede controlar sistemas robustos. Tiene un elevado ancho de banda, es fácil de diseñar.	Es limitado ante sistemas no lineales. No es bueno para diseños complejos.
PID EN CASCADA	Es estable en la operación, tiene menor margen de error.	Sólo es posible aplicarlo cuando se conoce la variación del sistema (perturbación).
PID DISCRETO	Tiene un comportamiento preciso, tiene la capacidad de implementar algoritmos complejos.	Dificultad de diseño y requiere un cierto nivel para el procesamiento.
DIFUSO	Ofrece salidas de una forma veloz y precisa, es adecuado para sistemas no lineales.	Presenta desventajas ante sistemas lineales debido a que no presta modelos matemáticos.
PID-FUZZY	Es bueno ante sistemas lineales y no lineales, es estable ante ciertos estímulos.	Requiere mayor nivel de procesamiento.

Fuente: Elaboración propia

## Capítulo 3

# MODELO MATEMÁTICO

### 3.1. Locomoción robot ápodo

Al observar a los robots del tipo ápodo la primera impresión que genera es su bioinspiración debido a su movimiento sinusoidal característico de animales sin extremidades, como por ejemplo las serpientes o gusanos. Además, se pueden ver sus similitudes de movimiento del robot serpiente diseñado por Miller [11] con el de la serpiente, mostrados en la (*Figura 3.1*). Gracias a esa forma de desplazamiento los robots ápodos pueden movilizarse en lugares estrechos de difícil acceso.

Figura 3.1: El robot Miller y serpiente



Fuente: Juan González Gómez [11]

El movimiento sinusoidal genera un desplazamiento en dos dimensiones (2D) en el plano, que al modificar su configuración combinando dos módulos con desfase de 90 grados generan un desplazamiento en tres dimensiones (3D) como es expuesto por

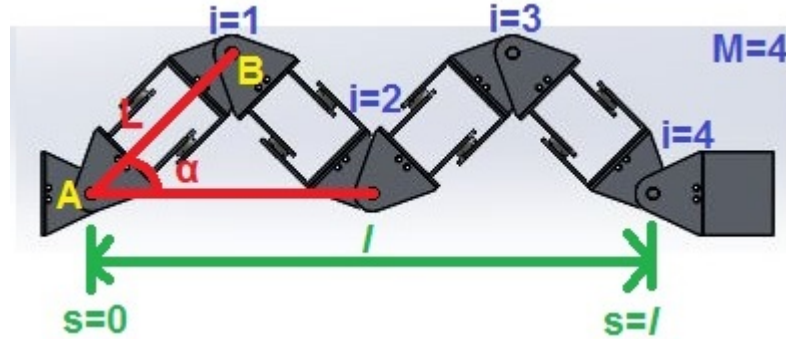
Andrés San Millán Rodríguez [23], usando como argumento la transmisión de ondas sinusoidales para la generación de curvas serpentínoides que mejoran su adaptación al entorno.

La curva generada por el movimiento característico del robot está dada como la variación sinusoidal del ángulo con base a la distancia a lo largo de la curva, siendo necesario analizar los siguientes parámetros:

- **Ángulo de serpenteo ( $\alpha$ ):** Es el ángulo generado entre los segmentos del robot y un eje de referencia en el plano donde se refleja la onda sinusoidal.
- **Ángulo de desfase ( $\phi$ ):** Comprende el ángulo de desplazamiento entre la onda sinusoidal y el punto de referencia.
- **Número de módulos ( $M$ ):** Los módulos son la estructura principal que conforma el cuerpo del robot ápedo, el número necesario de módulos mínimo para la conformación de la estructura del robot es  $M \geq 2$  y el máximo será limitado según la tarea asignada por el usuario al variar el número de cables, batería, sensores entre otros.
- **Longitud del módulo ( $L$ ):** La longitud  $L$  está dada por la medida del módulo cuando está en reposo, es decir que el ángulo es  $0^\circ$ . Debido a que los segmentos son iguales, cada uno estará compuesto por 2 módulos ( $2L$ ).
- **Longitud del robot ( $l$ ):** Al ser construido por módulos iguales la longitud del robot se encontrará a través del producto entre la longitud de los módulos ( $L$ ) con el número de estos ( $M$ ) por lo que se obtiene la ecuación  $l = M \times L$ .
- **Distancia a lo largo de la curva ( $s$ ):** Es la distancia entre 0 y  $l$  a lo largo de la curva.
- **Número de ondulación ( $k$ ):** El número de ondulaciones es la cantidad de repeticiones de la onda sinusoidal con base a la longitud del robot.
- **Periodo ( $T$ ):** Es el tiempo tomado en realizar una ondulación. Tomando en cuenta los parámetros de longitud del robot ( $l$ ) y número de ondulaciones ( $k$ ) mencionados anteriormente se puede generar la ecuación  $T = \frac{l}{k}$ .
- **Número de articulación ( $i$ ):** Comprende el número de la articulación, del 1 hasta  $M$ .

En la (Figura 3.2) se expone un plano del robot ápedo para su análisis con base a los parámetros ya mencionados anteriormente, debido al movimiento sinusoidal se entiende en estas dimensiones.

Figura 3.2: Gráfica de parámetros



Fuente: Elaboración propia

Como se expone en la figura anterior (*Figura 3.2*) los parámetros varían con base a la estructura, un ejemplo de eso es la longitud del módulo ( $L$ ) encontrada entre la distancia de A y B en la gráfica. Esa sección está compuesta por dos módulos con los cuales proporciona su desplazamiento en el plano de estudio, gracias a eso el número de módulos para su análisis es 4, la longitud del desplazamiento según la onda de serpenteo es dado por  $s$  y su longitud máxima en una repetición empleando la trayectoria planeada es de  $l$  con base a la gráfica.

La ecuación de la curva resultante que está dada con base a los parámetros mencionados anteriormente es:

$$K_{(s)} = -\frac{2\Pi k}{l}\alpha \sin\left(\frac{2\Pi k}{l}s\right) \quad (3.1)$$

Gracias a la curva definida es necesario tomar en cuenta un ángulo de desfase simbolizado por  $\phi$ , debido a eso la ecuación 3.1 se altera dando como resultado la siguiente ecuación:

$$K_{(s,\phi)} = -\frac{2\Pi k}{l}\alpha \sin\left(\phi + \frac{2\Pi k}{l}s\right) \quad (3.2)$$

La curva generada por  $K_{(s,\phi)}$  está definida como el cambio del vector tangente a la curva, por lo que se puede expresar como:

$$K_{(s,\phi)} = \frac{d\alpha_s}{ds} \quad (3.3)$$

Por lo que la resultante del ángulo que forma la tangente a la curva del serpenteo es:

$$\alpha_s = \alpha \cos\left(\phi + \frac{2\Pi k}{l}s\right) \quad (3.4)$$

Donde el ángulo de serpenteo ( $\alpha$ ) es formado por la tangente de la curva que pasa por el punto  $s=0$  definiendo la forma de la curva de la onda.

Las coordenadas de la onda en el plano cartesiano se encuentran entre X y Y se calculan mediante la integración del coseno y seno respectivamente de la ecuación 3.4 entre 0 y  $s$  obteniendo como resultado:

$$X_{(s,\phi)} = \int \cos \left( \alpha \cos \left( \phi + \frac{2\pi k}{l} s \right) \right) ds \quad (3.5)$$

$$Y_{(s,\phi)} = \int \sin \left( \alpha \cos \left( \phi + \frac{2\pi k}{l} s \right) \right) ds \quad (3.6)$$

El ángulo mínimo de serpenteo está dado por la ecuación 3.7 obtenida de [10], debido al cambio del diseño del robot ápodico empleado como objeto de estudio y el ángulo de serpenteo máximo está dada por la ecuación 3.8 que no puede sobrepasar el ángulo de  $121^\circ$ . En dado caso que se sobrepasara el ángulo máximo ocasionaría un colapso entre los puntos de la curva, por lo que se debe tener en cuenta esta restricción.

$$\alpha_{\text{mínimo}} \approx \frac{121}{L} \left( \sqrt{\left( \frac{l}{L} \right)^2 + 18} - \frac{l}{L} \right) \quad (3.7)$$

$$\alpha_{\text{máximo}} = \frac{A}{2 \sin \left( \frac{\pi * k}{M} \right)} \leq 121 \quad (3.8)$$

Aplicando las ecuaciones 3.5 y 3.6 vistas anteriormente en el plano cartesiano se puede visualizar la forma de la onda de serpenteo adquirida a través de los parámetros dados por el modelo mecánico del robot ápodico, por lo tanto, se tienen en cuenta la longitud total ( $l$ ) dada por el número de módulos ( $M$ ) y la longitud de cada uno ( $L$ ), ocasionando que sean variables fijas con base al robot. Así como hay variables dependientes del robot, también hay variables que el usuario o las circunstancias dadas en el campo de aplicación como son: el ángulo de serpenteo ( $\alpha$ ) y el número de oscilaciones ( $k$ ). Algunos resultados de la ecuación de serpenteo con base a las ecuaciones 3.5 y 3.6 usando algunas variaciones del movimiento se muestran en las siguientes figuras utilizando la interfaz gráfica *Matlab*® implementando el pseudocódigo mostrado en el algoritmo 3.1.



**Algoritmo 3.1** Serpenteo**Programa:** Gráfica de serpenteo**Entradas:**  $L, M, \alpha, k$ **Salida:**  $x, y$ **Variables:**  $s, d$ **Algoritmo** $l = L * M$ Desde  $s = 1$  hasta  $l$  $d =$  deriva la ecuación 3.4 $x =$  Integral  $\cos(d)$  desde 0 hasta  $s$  $y =$  Integral  $\sin(d)$  desde 0 hasta  $s$ 

sobrepone la imagen

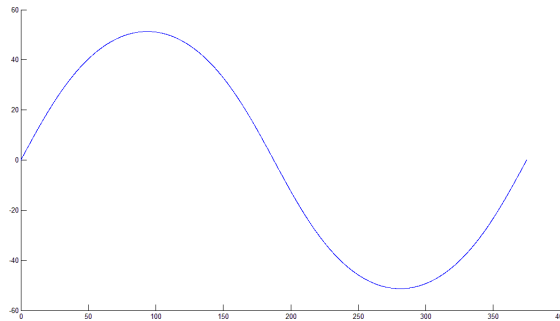
imprime coordenadas  $(x, y)$ 

Siguiente

Fin del programa

En la (Figura 3.3) se observa el movimiento de serpenteo con ocho módulos ( $M=8$ ), una ondulación de movimiento ( $k=1$ ) y cuarenta y cinco grados de ángulo de serpenteo  $\alpha = 45^\circ$ :

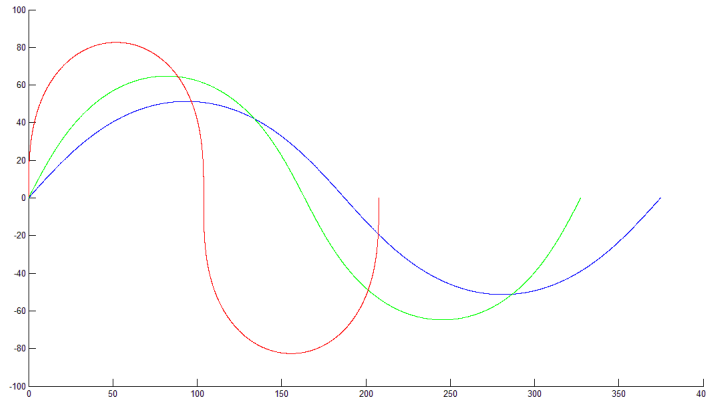
Figura 3.3: Onda de serpenteo con curvatura de  $45^\circ$  en *Matlab*®.



Fuente: Elaboración propia

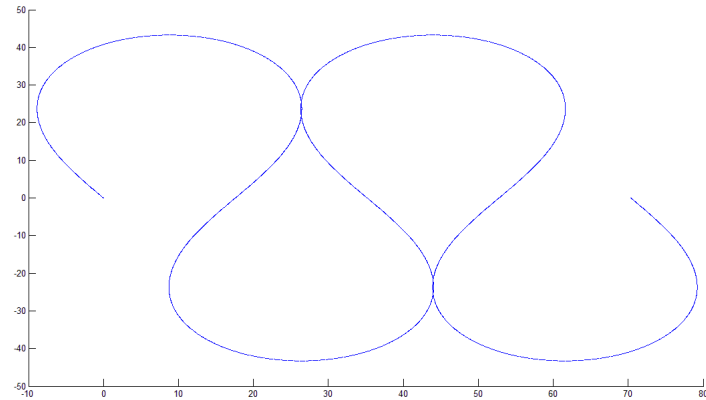
Para ver las diferencias e importancia en los ángulos de serpenteo en la (Figura 3.4) se observan tres ondas sinusoidales, en la que noventa grados ( $90^\circ$ ) es de color rojo, sesenta grados ( $60^\circ$ ) de color verde y cuarenta y cinco grados ( $45^\circ$ ) de color azul.

Figura 3.4: Variación de ángulo de serpenteo



Fuente: Elaboración propia

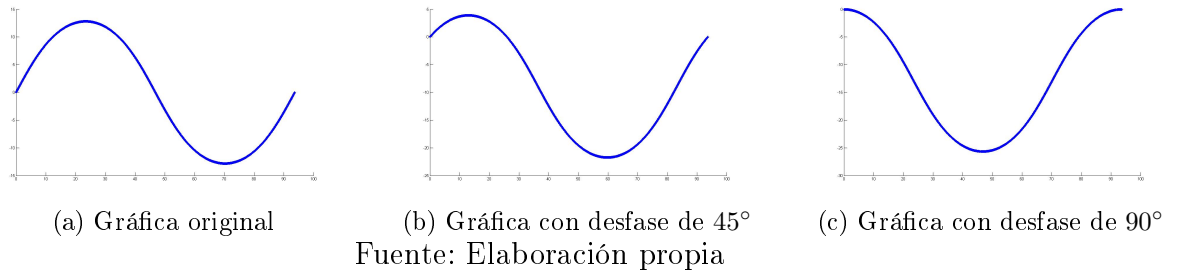
El ángulo máximo que pueden adoptar los robots ápodos es de ciento veintiún grados ( $121^\circ$ ) mostrado en la (*Figura 3.5*) exponiendo una curvatura muy pronunciada, gracias a eso fue necesario utilizar dos ondulaciones ( $k=2$ ) para poder apreciar su forma.

Figura 3.5: Onda de serpenteo con curvatura de  $121^\circ$  en *Matlab*®

Fuente: Elaboración propia

Los desfases ( $\phi$ ) en las ecuaciones 3.5 y 3.6 muestran alteraciones en la gráfica como se visualiza en la (*Figura 3.6*), donde se muestra la gráfica original obtenida con el ángulo de serpenteo de  $45^\circ$  junto con las gráficas obtenidas con desfase de cuarenta y cinco grados ( $\phi = 45^\circ$ ) y desfase de noventa grados ( $\phi = 90^\circ$ ).

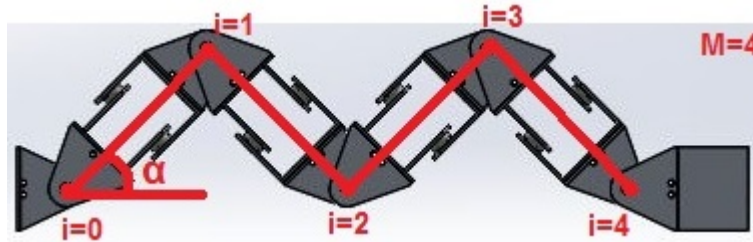
Figura 3.6: Desfase en la curva de serpenteo



### 3.2. Modelado del robot ápodo

Para poder realizar el modelado matemático de forma adecuada es necesario implementar el caso discreto en el cual el ángulo de serpenteo cambia en función de la variable  $i$  al indicar el número de articulaciones que varían entre 1 hasta  $M$  como se muestra en la (Figura 3.7).

Figura 3.7: Gráfica en caso discreto



Fuente: Elaboración propia

En la (Figura 3.7) vista anteriormente se expone la naturaleza discreta del robot ápodo. Gracias a eso se puede entender la razón de cambio entre los módulos en el plano discreto, considerando que la forma de la curva de serpenteo en ese plano cambia conforme a la variable discreta  $i$  mencionada entre los parámetros en vez de emplear la variable continua  $s$  que era comprendida en el espacio continuo como la distancia longitudinal del cuerpo del robot resultante del producto del número de módulos ( $M$ ) y la longitud de cada uno ( $L$ ).

Al realizar el análisis correspondiente en el plano discreto se emplea la ecuación 3.4 de forma discreta con base a las variables  $M$  (Número de módulos) y su número de articulaciones  $i$  dando como resultado la siguiente ecuación :

$$\alpha_i = \alpha \cos \left( \phi + \frac{2\pi k}{M}(i - 1) \right) \quad (3.9)$$

La ecuación 3.9, al representar el ángulo de la onda formada en el plano x-y, ocasionando que las coordenadas cartesianas de las articulaciones sean definidas con las ecuaciones 3.10 y 3.11, cuya configuración en funciones trigonométricas contemplan la posición del módulo en el pasado para localizarla en el momento, por lo que se debe realizar la suma de su posición hasta  $M$  módulos como se muestra las siguientes ecuaciones:

$$X_{(i)} = \begin{cases} \frac{L}{2} \cos \alpha & i = 1 \\ X_{(i-1)} + L \cos \alpha_i & i \in [2, M] \end{cases} \quad (3.10)$$

$$Y_{(i)} = \begin{cases} \frac{L}{2} \sin \alpha & i = 1 \\ Y_{(i-1)} + L \sin \alpha_i & i \in [2, M] \end{cases} \quad (3.11)$$

Utilizando las ecuaciones 3.10 y 3.11 expuestas anteriormente, se visualizan en el plano cartesiano las posiciones de los módulos adquiriendo la forma de la onda de serpenteo estudiada gracias a los parámetros proporcionados por el modelo mecánico del robot ápedo, por ende se debe tener en cuenta el número de módulos ( $M$ ) y la longitud de cada uno ( $L$ ) ya que son variables fijas con base al robot. Así como hay variables dependientes del robot, también hay variables que el usuario o las circunstancias dadas en el campo de aplicación, como son: el ángulo de serpenteo ( $\alpha$ ) y el número de oscilaciones ( $k$ ). Alguno de los resultados de la ecuaciones discretas de serpenteo (ecuaciones 3.10 y 3.11) haciendo algunas variaciones se exhiben a continuación utilizando la interfaz gráfica *Matlab®* implementando el pseudocódigo mostrado en el algoritmo 3.2.

En la (*Figura 3.8*) se expone la posición de los módulos para realizar el movimiento de serpenteo con ondulación de movimiento ( $k=1$ ) y cuarenta y cinco grados de ángulo de serpenteo  $\alpha = 45^\circ$ , pero con diferentes números de módulos. La gráfica posicionada al lado izquierdo implementa dieciséis módulos ( $M=16$ ) logrando una igualdad considerable con la (*Figura 3.3*), y en la gráfica del lado derecho expone la localización de ocho módulos ( $M=8$ ) el cual adquiere la posición para realizar el movimiento de serpenteo limitada por los módulos ocasionando una forma más forzada para la localización de los módulos.

La diferencia de los ángulos de serpenteo expuestos en la (*Figura 3.9*) donde la gráfica de color rojo es de noventa grados ( $90^\circ$ ) y es de color rojo, la gráfica verde es de sesenta grados ( $60^\circ$ ) y la gráfica azul es de cuarenta y cinco grados ( $45^\circ$ ). Similar a los presentados anteriormente en la (*Figura 3.4*) que describe el movimiento, a diferencia de la (*Figura 3.9*) al describir la posición del robot.

**Algoritmo 3.2** Serpenteo en caso discreto**Programa:** Gráfica de serpenteo en caso discreto**Entradas:**  $L, M, \alpha, k$ **Salida:** Vector de las coordenadas en X, vector de las coordenadas en Y**Variables:** i**Algoritmo**

X= matriz vacía

Y= matriz vacía

Desde i = 1 hasta M

Si i = 1

$$x = \frac{L}{2} \cos \alpha$$

$$y = \frac{L}{2} \sin \alpha$$

X=[coordenada inicial (0) X x]

Y=[coordenada inicial (0) Y y]

Fin Si

Si i &gt; 1

$$x = x + L \cos \alpha$$

$$y = y + L \sin \alpha$$

X=[coordenada inicial (0) X x]

Y=[coordenada inicial (0) Y y]

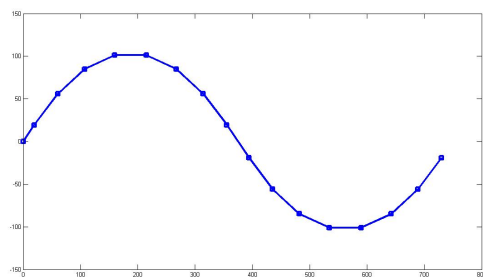
Fin Si

Siguiente

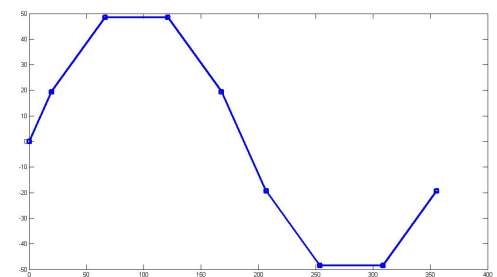
Imprime vectores (X,Y)

Fin del programa

Figura 3.8: Posición de los módulos



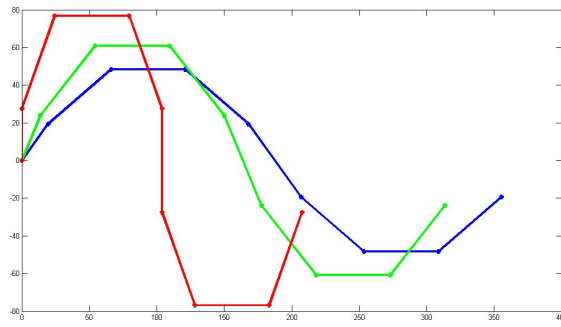
(a) Posición 16 módulos



(b) Posición 8 módulos

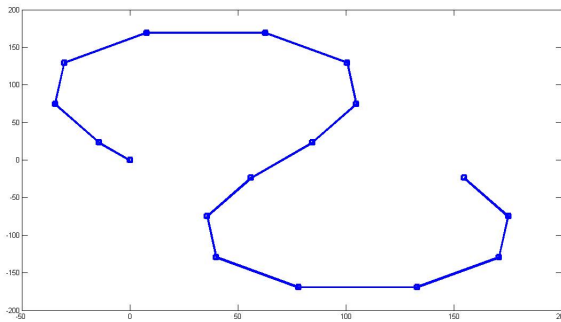
Fuente: Elaboración propia

Figura 3.9: Variación de ángulo de serpenteo en el plano discreto



Fuente: Elaboración propia

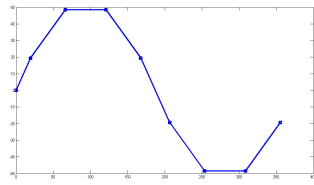
Aplicando el ángulo máximo de serpenteo de ciento veintiún grados ( $121^\circ$ ) se obtiene la (Figura 3.10), exponiendo la implementación de dieciséis módulos ( $M=16$ ) para obtener una sola oscilación ( $k=1$ ) para alcanzar su forma adecuada.

Figura 3.10: Onda de serpenteo con curvatura de 121 grados en el plano discreto en *Matlab*®.

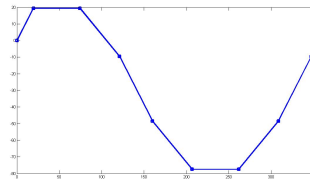
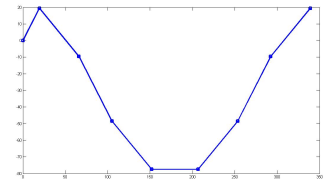
Fuente: Elaboración propia

Evalutando los desfases en torno a las ecuaciones 3.10 y 3.11, es encontrada la (Figura 3.11) donde son exhibidas las gráficas de desfase de cuarenta y cinco grados ( $\phi = 45^\circ$ ) y de noventa grados ( $\phi = 90^\circ$ ) junto a la gráfica original, destacando el punto de inicio de las gráficas en las coordenadas  $(0,0)$  al estar basados en los vectores de posición obtenido del movimiento de serpenteo en el espacio discreto.

Figura 3.11: Desfase en la curva de serpenteo



(a) Gráfica original

(b) Gráfica con desfase de  $45^\circ$ (c) Gráfica con desfase de  $90^\circ$ 

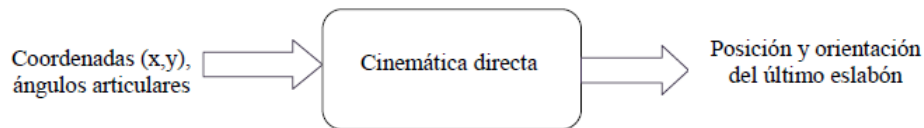
Fuente: Elaboración propia

### 3.3. Cinemática directa

La cinemática directa busca determinar la posición y orientación del último eslabón de la cadena cinemática, como es expuesto por Jessica Hazdey Barreto Paz y Lizet Tatiana Medina Ramírez [1], a través de su diagrama (*Figura 3.12*) expone que empleando el sistema de coordenadas en los ángulos de articulaciones e implementando la cinemática directa se obtiene la posición y orientación del ultimo eslabón. Para realizar esto se utiliza una transformada homogénea. Con el objetivo de realizar la transformada homogénea, el algoritmo se divide en tres fases:

1. Definición de los parámetros de la matriz Denavit Hartenberg.
2. Asignación de los sistemas de referencia.
3. Cálculo de la transformada homogénea.

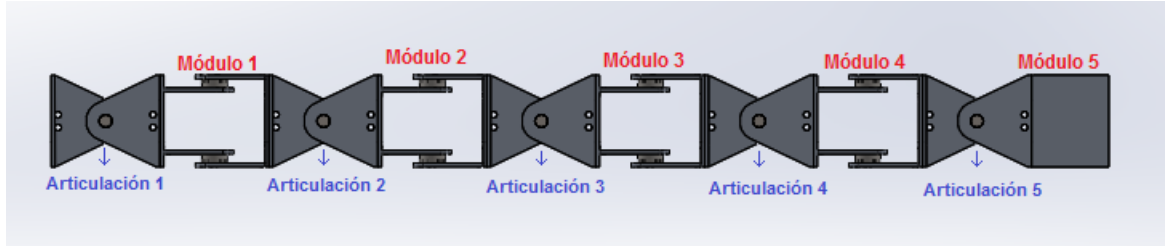
Figura 3.12: Parámetros de la cinemática directa



Fuente: Jessica Hazdey Barreto Paz y Lizet Tatiana Medina Ramírez [1]

Para realizar la primera fase se utilizará como base el diseño de robot ápodo mostrado en la (*Figura 3.13*) limitando su locomoción al plano xy para simplificar su estudio y empleando como referencia el último módulo para realizar la matriz de Denavit – Hartenberg sin olvidar que el robot cuenta con cinco articulaciones y cinco módulos en ese plano.

Figura 3.13: Modelo de robot ápedo



Fuente: Elaboración propia

Al analizar los grados de libertad que posee el robot se logró definir los parámetros mostrados en el cuadro 3.1 donde  $L$  es la longitud del módulo y considerando que en reposo adquiere la forma de línea recta.

Cuadro 3.1: Parámetros Denavit-Hartenberg del robot ápedo.

Articulación	$\theta_i$	$A_i$	$D_i$	$\alpha_i$
1	0	$L/2$	0	$\alpha_1$
2	0	$L$	0	$\alpha_2$
3	0	$L$	0	$\alpha_3$
4	0	$L$	0	$\alpha_4$
5	0	$L$	0	$\alpha_5$

Fuente: Elaboración propia

Para realizar la segunda fase es necesario asignar los sistemas de referencia, las transformaciones homogéneas obtenidas para cada articulación son las siguientes:

$$A_1 = \begin{bmatrix} \cos \alpha_1 & -\sin \alpha_1 & 0 & \frac{L}{2} * \cos \alpha_1 \\ \sin \alpha_1 & \cos \alpha_1 & 0 & \frac{L}{2} * \sin \alpha_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

$$A_2 = \begin{bmatrix} \cos \alpha_2 & -\sin \alpha_2 & 0 & L * \cos \alpha_2 \\ \sin \alpha_2 & \cos \alpha_2 & 0 & L * \sin \alpha_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

$$A_3 = \begin{bmatrix} \cos \alpha_3 & -\sin \alpha_3 & 0 & L * \cos \alpha_3 \\ \sin \alpha_3 & \cos \alpha_3 & 0 & L * \sin \alpha_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

$$A_4 = \begin{bmatrix} \cos \alpha_4 & -\sin \alpha_4 & 0 & L * \cos \alpha_4 \\ \sin \alpha_4 & \cos \alpha_4 & 0 & L * \sin \alpha_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$



$$A_5 = \begin{bmatrix} \cos \alpha_5 & -\sin \alpha_5 & 0 & L * \cos \alpha_5 \\ \sin \alpha_5 & \cos \alpha_5 & 0 & L * \sin \alpha_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

Para llevar a cabo la tercera fase, se realiza el cálculo de la transformada homogénea con el fin de ver la orientación y posición del último módulo. Para realizar dicho cálculo, se hace la multiplicación de las transformadas desde la 3.12 hasta la 3.16, es decir,  $T = A_1 * A_2 * A_3 * A_4 * A_5$ . La transformada homogénea resultante se observa en la 3.17.

$$T = \begin{bmatrix} \cos \alpha_i & -\sin \alpha_i & 0 & \frac{L}{2} \cos \alpha_1 + L \cos (\alpha_1 + \alpha_2) + L \cos (\alpha_1 + \alpha_2 + \alpha_3) + L \cos \alpha_i \\ \sin \alpha_i & \cos \alpha_i & 0 & \frac{L}{2} \sin \alpha_1 + L \sin (\alpha_1 + \alpha_2) + L \sin (\alpha_1 + \alpha_2 + \alpha_3) + L \sin \alpha_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

Donde  $\alpha_i$  se determina por la sumatoria de  $\alpha$  dependiendo del número de módulos que se analizan, por lo tanto,  $\alpha_i = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5$ . Además como se observa en la (Figura 3.13), la posición inicial del robot es totalmente recto, es decir,  $\theta_i = d_i = 0$ , debido a que no tiene movimiento rotacional ni se encuentra en el espacio comprendido en la matriz de Denavit Hartenberg y considerando que  $L$  es la longitud del eslabón en el espacio de la matriz en  $A$ . Luego de hallar la transformada homogénea resultante se hallan las posiciones finales representadas en las ecuaciones 3.18 y 3.19.

$$X_{final} = \frac{L}{2} \cos \alpha_1 + L \cos (\alpha_1 + \alpha_2) + L \cos (\alpha_1 + \alpha_2 + \alpha_3) + L \cos \alpha_i \quad (3.18)$$

$$Y_{final} = \frac{L}{2} \sin \alpha_1 + L \sin (\alpha_1 + \alpha_2) + L \sin (\alpha_1 + \alpha_2 + \alpha_3) + L \sin \alpha_i \quad (3.19)$$

### 3.3.1. Resultados cinemática directa

Los resultados de la cinemática directa son mostrados utilizando la herramienta gráfica *Matlab*® con el pseudocódigo visto en algoritmo 3.3, al facilitar su visualización y descripción de movimiento con base a las matrices y ecuaciones encontradas anteriormente.

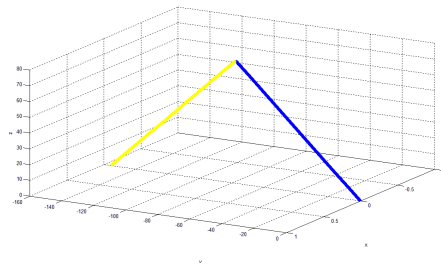
**Algoritmo 3.3** Cinemática Directa**Programa:** Cinemática directa**Entradas:** Matriz de Denavit Hartenberg**Salida:** Coordenadas X, Y**Variables:** theta, d, a, alpha**Algoritmo**

T=Multiplicación de la matriz de cada vector para encontrar las matriz homogénea.

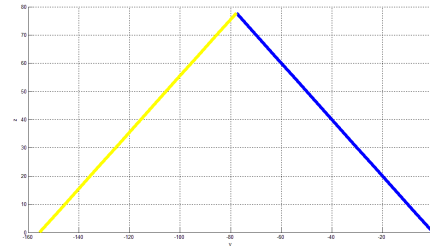
 $X = T(1, 4)$  $Y = T(2, 4)$ 

Fin del programa

Figura 3.14: Resultado de cinemática directa de un módulo



(a) Resultado de cinemática directa de dos módulos en 3D

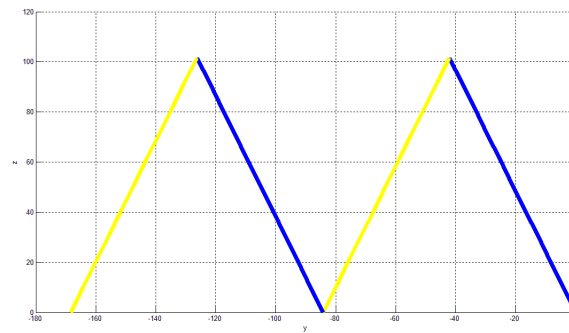


(b) Resultado de cinemática directa de dos módulos en 2D

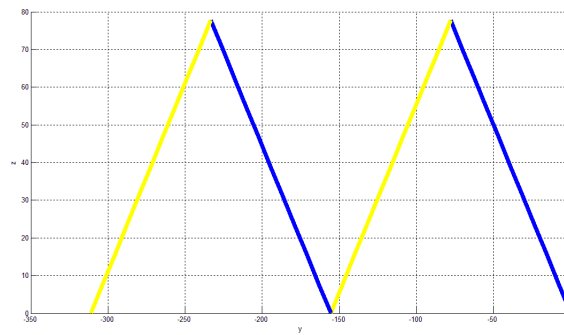
Fuente: Elaboración propia

En la (Figura 3.14) mostrada anteriormente se puede ver la representación gráfica de un módulo tanto en dos dimensiones (2D) como en tres dimensiones (3D), al mostrar un espacio donde se puede visualizar el movimiento del robot y estudiar su comportamiento con base a las limitaciones físicas en la que se basa el robot.

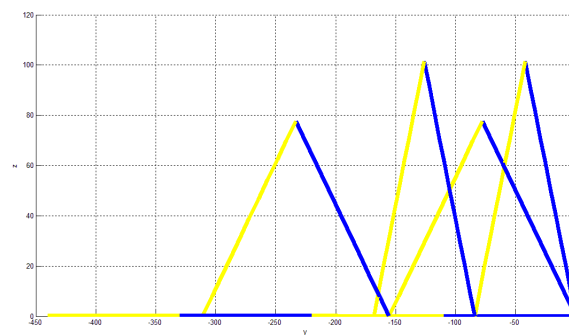
Figura 3.15: Resultado de cinemática directa



(a) Resultado de cinemática directa cuatro módulos con alpha de 90 grados



(b) Resultado de cinemática directa cuatro módulos con alpha de 135 grados



(c) Resultado de cinemática directa comparación de movimiento

Fuente: Elaboración propia

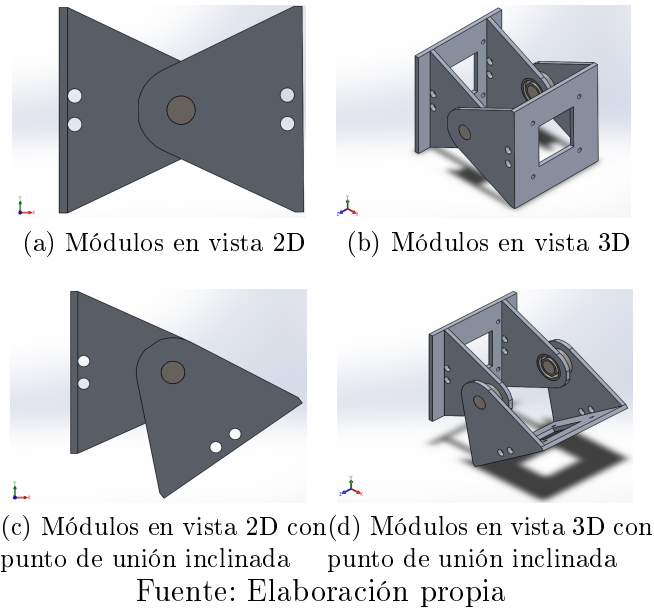
La variación de movimiento mostrada en la (Figura 3.15) se realiza con base a la variación de alpha que representa el ángulo formado entre los módulos que componen la estructura del robot usado como objeto de estudio, en la que se compara los cambios del

ángulo  $\alpha$  comprendido entre  $90^\circ$  y  $180^\circ$  alcanzando la posición del robot en reposo, realizando un desplazamiento notable al ser analizado con base a una estructura de cuatro módulos.

### 3.4. Cinemática inversa

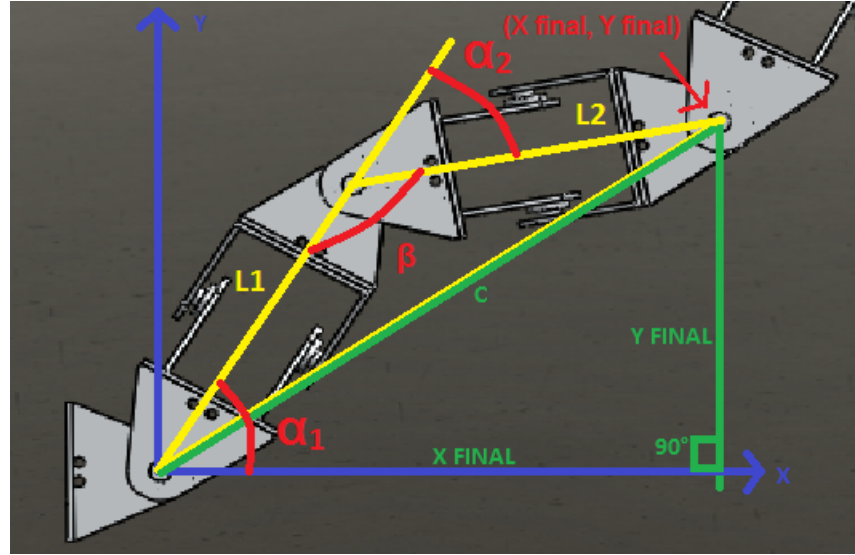
La cinemática inversa a diferencia de la cinemática directa emplea la posición inicial del módulo junto a la posición final del módulo con base a las coordenadas cartesianas en el eje  $xy$  como objeto de estudio. Con ese objetivo, se empieza analizando el movimiento entre dos módulos expuesto en la (Figura 3.16) debido a que el ángulo de inclinación generado entre los módulos indica la posición del siguiente.

Figura 3.16: Parámetros de movimiento entre los módulos



Al emplear el concepto mostrado anteriormente se encuentra una relación geométrica entre los módulos como se muestran en la (Figura 3.17), al exponer el cambio de posición con base a los ángulos de  $\alpha_1$  y  $\alpha_2$  considerando los parámetros expuestos en este capítulo, principalmente la longitud del módulo  $L$  debido a que es uno de los factores principales para encontrar la posición y orientación final de la articulación usando la ley trigonométrica de «La ley del coseno».

Figura 3.17: Posición entre módulos



Fuente: Elaboración propia

La ley del coseno debe su importancia a que facilita establecer una relación de posición con base al triángulo, teniendo en cuenta la interacción entre dos lados del triángulo y el ángulo formado entre esos lados, estas variables se ven reflejadas claramente en la (Figura 3.17) exponiendo los parámetros que forman el triángulo oblicuo junto a las medidas de un triángulo rectángulo formando por las coordenadas cartesianas  $X_{final}$  y  $Y_{final}$  correspondientes a los catetos adyacentes y una hipotenusa formada por la longitud comprendida entre los puntos del plano  $(X_{final}, Y_{final})$  y  $(0,0)$ , teniendo en cuenta que la longitud entre esos dos puntos está dada por el teorema de Pitágoras. Las ecuaciones resultantes de la ley del coseno es dada por la ecuación 3.20 y el teorema de Pitágoras encontrado en la ecuación 3.21.

$$c^2 = a^2 + b^2 - 2ab \cos \beta \quad (3.20)$$

$$c^2 = X_{final}^2 + Y_{final}^2 \quad (3.21)$$

Al tener en consideración que los valores de « $c^2$ » en las ecuaciones 3.20 y 3.21 es posible igualar esas dos ecuaciones y reemplazando  $a$  y  $b$  es decir  $L_1$  y  $L_2$  y utilizando relaciones trigonométricas se encuentra que  $\beta = 180 - \alpha_2$  obteniendo ecuación 3.22.

$$X_{final}^2 + Y_{final}^2 = L_1^2 + L_2^2 - 2L_1L_2 \cos (180 - \alpha_2) \quad (3.22)$$

Despejando el ángulo de  $\alpha_2$  se encuentra la ecuación 3.23.

$$\alpha_2 = \arccos \left( \frac{X_{final}^2 + Y_{final}^2 - (L_1)^2 - (L_2)^2}{2L_1L_2} \right) \quad (3.23)$$

Con el objetivo de hallar la relación trigonométrica representativa del ángulo  $\alpha_1$  se obtuvo que:

$$X_{final} = L_1 \cos(\alpha_1) + L_2 \cos(\alpha_1 + \alpha_2) \quad (3.24)$$

$$Y_{final} = L_1 \sin(\alpha_1) + L_2 \sin(\alpha_1 + \alpha_2) \quad (3.25)$$

Buscando encontrar la distancia entre el origen y el extremo de los puntos de referencia se obtiene que:

$$X_{final}^2 + Y_{final}^2 = (L_1 \cos(\alpha_1) + L_2 \cos(\alpha_1 + \alpha_2))^2 + (L_1 \sin(\alpha_1) + L_2 \sin(\alpha_1 + \alpha_2))^2 \quad (3.26)$$

Al resolver los cuadrados de la ecuación 3.26 se obtiene:

$$X_{final}^2 + Y_{final}^2 = A + B \quad (3.27)$$

$$A = (L_1^2 \cos^2(\alpha_1) + L_2^2 \cos^2(\alpha_1 + \alpha_2) + 2L_1 L_2 \cos(\alpha_1) \cos(\alpha_1 + \alpha_2)) \quad (3.28)$$

$$B = (L_1^2 \sin^2(\alpha_1) + L_2^2 \sin^2(\alpha_1 + \alpha_2) + 2L_1 L_2 \sin(\alpha_1) \sin(\alpha_1 + \alpha_2)) \quad (3.29)$$

Simplificando la ecuación 3.27 se encuentra que:

$$X_{final}^2 + Y_{final}^2 = L_1^2 + L_2^2 + 2L_1 L_2 (\cos(\alpha_1) \cos(\alpha_1 + \alpha_2) + \sin(\alpha_1) \sin(\alpha_1 + \alpha_2)) \quad (3.30)$$

Considerando las propiedades trigonométricas expuestas en las ecuaciones 3.31 y 3.32 se obtuvo la ecuación 3.33 que representa la distancia entre el punto de origen y el extremo, mostradas a continuación:

$$\cos(a \pm b) = \cos(a) \cos(b) \pm \sin(a) \sin(b) \quad (3.31)$$

$$\sin(a \pm b) = \cos(a) \sin(b) \pm \cos(a) \sin(b) \quad (3.32)$$

$$X_{final}^2 + Y_{final}^2 = L_1^2 + L_2^2 + 2L_1 L_2 \cos(\alpha_2) \quad (3.33)$$

Para buscar el valor de  $\alpha_1$  se emplean las ecuaciones ya vistas de  $X_{final}$  y  $Y_{final}$ , expuestas a continuación:

$$X_{final} = L_1 \cos(\alpha_1) + L_2 \cos(\alpha_1 + \alpha_2) \quad (3.34)$$

$$X_{final} = L_1 \cos(\alpha_1) + L_2 \cos(\alpha_1) \cos(\alpha_2) - L_2 \sin(\alpha_1) \sin(\alpha_2) \quad (3.35)$$

$$X_{final} = \cos(\alpha_1)(L_1 + L_2 \cos(\alpha_2)) - \sin(\alpha_1)(L_2 \sin(\alpha_2)) \quad (3.36)$$

$$Y_{final} = L_1 \sin(\alpha_1) + L_2 \sin(\alpha_1 + \alpha_2) \quad (3.37)$$

$$Y_{final} = L_1 \sin(\alpha_1) + L_2 \sin(\alpha_1) \cos(\alpha_2) + L_2 \sin(\alpha_2) \cos(\alpha_1) \quad (3.38)$$

$$Y_{final} = \cos(\alpha_1)(L_2 \sin(\alpha_2)) + \sin(\alpha_1)(L_1 + L_2 \cos(\alpha_2)) \quad (3.39)$$

Con el objetivo de encontrar  $\alpha_1$  se generaron las ecuaciones 3.36 y 3.39 que poseen dos incógnitas por lo que se pueden resolver de la siguiente manera:

$$\cos(\alpha_1) = \frac{X_{final} + (\sin \alpha_1)(L_2 \sin(\alpha_2))}{(L_1 + L_2 \cos(\alpha_2))} \quad (3.40)$$

$$Y_{final} = \frac{X_{final} + (\sin \alpha_1)(L_2 \sin(\alpha_2))}{(L_1 + L_2 \cos(\alpha_2))} (L_2 \sin(\alpha_2)) + \sin(\alpha_1)(L_1 + L_2 \cos(\alpha_2)) \quad (3.41)$$

$$Y_{final} = \frac{X_{final} L_2 \sin(\alpha_2) + \sin(\alpha_1)(L_1^2 + L_2^2 + 2L_1 L_2 \cos(\alpha_2))}{(L_1 + L_2 \cos(\alpha_2))} \quad (3.42)$$

$$\sin(\alpha_1) = \frac{Y_{final}(L_1 + L_2 \cos(\alpha_2)) - X_{final} L_2 \sin(\alpha_2)}{X_{final}^2 + Y_{final}^2} \quad (3.43)$$

Con base a las anteriores ecuaciones finalmente se despeja  $\alpha_1$  de la ecuación 3.43 dando como resultado la ecuación 3.44:

$$\alpha_1 = \arcsin \left( \frac{Y_{final}(L_1 + L_2 \cos(\alpha_2)) - X_{final} L_2 \sin(\alpha_2)}{X_{final}^2 + Y_{final}^2} \right) \quad (3.44)$$

Considerando las ecuación 3.23 perteneciente al ángulo de  $\alpha_2$  y la ecuación 3.44 perteneciente al ángulo de  $\alpha_1$  son encontradas las posibles singularidades, las cuales son causadas por las limitaciones físicas presentadas por el robot objeto de estudio y las ecuaciones de la cinemática inversa al no contemplar sus limitaciones, por lo que deben ser tenidas en cuenta las dimensiones para no plantear coordenadas inadecuadas para el sistema tanto por su lejanía como su cercanía entorno al robot.

### 3.4.1. Resultado de la cinemática inversa

La visualización de los resultados de la cinemática inversa utilizando las ecuaciones 3.23 y 3.44 anteriormente vistas fueron realizadas gracias a la herramienta gráfica de *Matlab*<sup>®</sup> utilizando el pseudocódigo 3.4 permitiendo realizar los cálculos correspondientes a la descripción de movimiento.

---

**Algoritmo 3.4** Cinemática Inversa
 

---

**Programa:** Cinemática inversa

**Entradas:** Coordenada finales X y Y

**Salida:** Ángulo  $\alpha_1$  y  $\alpha_2$

**Variables:**  $X_{final}$  y  $Y_{final}$

**Algoritmo**

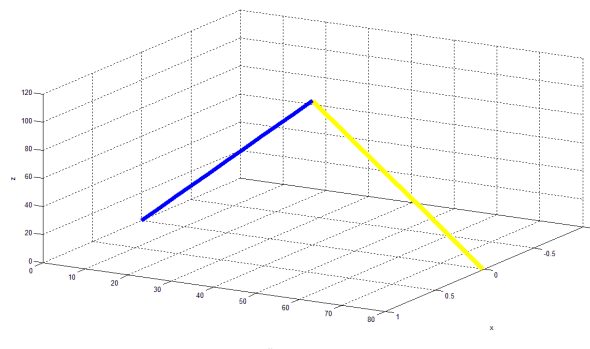
$\alpha_2 =$  ecuación 3.23

$\alpha_1 =$  ecuación 3.44

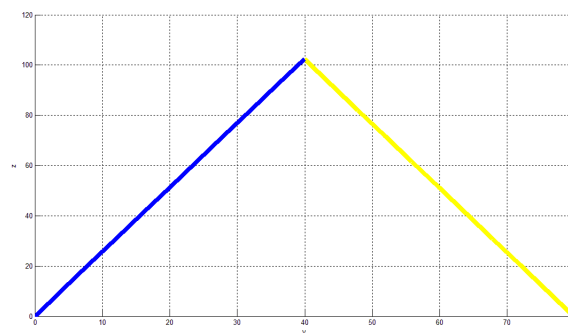
Fin del programa

---

Figura 3.18: Resultado de cinemática inversa de un módulo



(a) Resultado de cinemática inversa de dos módulos en 3D



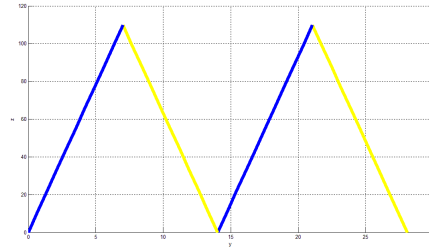
(b) Resultado de cinemática inversa de dos módulos en 2D

Fuente: Elaboración propia

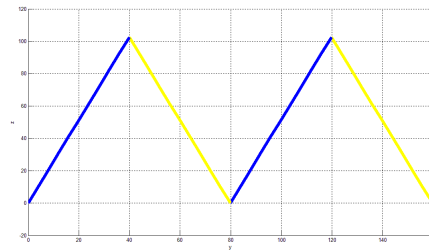


La (Figura 3.18) muestra la visualización de dos módulos tanto la representación en dos dimensiones (2D) como en tres dimensiones (3D), reflejando un cambio al basarse en la posición final y no en los ángulos requeridos para el posicionamiento del robot, esto es expuesto de forma sobresaliente en la (Figura 3.19).

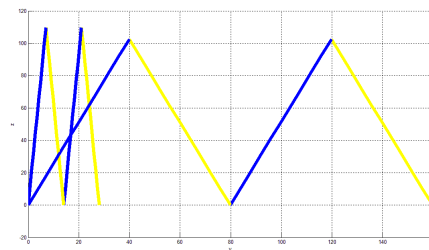
Figura 3.19: Resultado de cinemática inversa.



(a) Resultado de cinemática inversa cuatro módulos con coordenadas (27.5,0)



(b) Resultado de cinemática inversa cuatro módulos con coordenadas (360,0)



(c) Resultado de cinemática inversa comparación de movimiento

Fuente: Elaboración propia

Los resultados expuestos en la (Figura 3.19) reflejan los cambios al basarse en una posición final en el plano xy ocasionando una mayor precisión en el espacio. Los ángulos de referencia varían con base a la posición ocasionando que sea necesario un conocimiento de las propiedades del robot usado como objeto de estudio, debido a la facilidad de descontrol que podría ocasionar al contemplar un punto inalcanzable físicamente,

por lo tanto se requieren limitaciones fijadas por los parámetros de movimiento.

### 3.5. Velocidad angular

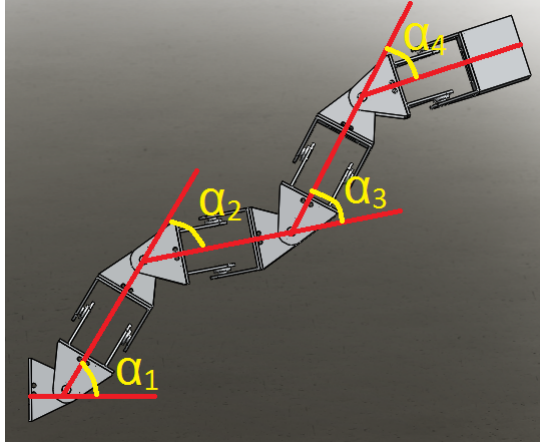
En la implementación de movimiento del robot ápodó surgen múltiples incógnitas para la comprensión adecuada del movimiento, en la que se destaca la velocidad angular del robot al ser un parámetro usado para calcular la velocidad de movimiento resultante para la configuración de diseño propia del robot. Por lo que es importante tener en cuenta la velocidad del extremo del robot, debido a que la velocidad de la articulación está dada por la velocidad del extremo del robot, como también en caso opuesto la velocidad del extremo del robot será dada por la velocidad de la articulación al estar relacionado de una forma directa.

Con el objetivo de encontrar la velocidad angular del motor se implementa la matriz jacobiana, con base a la posición del último módulo sabiendo que el vector de las velocidades de las articulaciones se expone en la ecuación 3.45.

$$\Theta = \begin{bmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \\ \dot{\alpha}_3 \\ \dot{\alpha}_4 \end{bmatrix} \quad (3.45)$$

Empleando como base la cinemática directa estudiada anteriormente, utilizando la configuración de cuatro eslabones mostrada en la (*Figura 3.20*) expuesto a continuación:

Figura 3.20: Robot ápodó de cuatro eslabones



Fuente: Elaboración propia

Con base a la anterior figura se encontró el vector de posición del extremo del robot visto en la ecuación 3.46 mostrado a continuación:

$$\begin{bmatrix} x_{final} \\ y_{final} \end{bmatrix} = \begin{bmatrix} \frac{L}{2} \cos \alpha_1 + L \cos (\alpha_1 + \alpha_2) + L \cos (\alpha_1 + \alpha_2 + \alpha_3) + L \cos (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) \\ \frac{L}{2} \sin \alpha_1 + L \sin (\alpha_1 + \alpha_2) + L \sin (\alpha_1 + \alpha_2 + \alpha_3) + L \sin (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) \end{bmatrix} \quad (3.46)$$

Dado que se está empleando con base a la cinemática directa, la posición del robot se encuentra en términos de los ángulos, por lo que se utiliza un vector en función « $f$ » de los ángulos articulados fundamentado en las coordenadas xy del extremo del robot de los ángulos usados como objeto de estudio, planteando la fórmula indicada en la ecuación 3.47:

$$\triangleq f(\Theta) = \begin{bmatrix} f_1(\Theta) \\ f_2(\Theta) \end{bmatrix} \quad (3.47)$$

Para obtener las velocidades de los motores, es necesario tener una velocidad deseada particular de cada robot, y es obtenida por la siguiente ecuación :

$$\dot{X}_{deseada} = \begin{bmatrix} \dot{x}_{deseada} \\ \dot{y}_{deseada} \end{bmatrix} \quad (3.48)$$

Utilizando la ecuación 3.48 se pueden determinar las velocidades de cada articulación a partir de una velocidad lineal y la matriz jacobiana de las coordenadas en función de los ángulos articulares, por lo que se encuentra la ecuación 3.49:

$$\dot{X}_{deseada} = \frac{d}{dt} f(\Theta) = \frac{\partial f(\Theta)}{\partial \Theta} \quad (3.49)$$

alcanzando la ecuación 3.50:

$$J(\Theta) = \frac{\partial f(\Theta)}{\partial \Theta} \quad (3.50)$$

En el cual en la ecuación 3.50 se encuentra la descripción de la matriz Jacobiana de la transformación no lineal de la función  $f$ . En el cual se obtiene una matriz con base a las derivadas de un vector respecto a otro vector. Aplicando este concepto a los eslabones enseñados en la (*Figura 3.20*) se obtiene la ecuación 3.59:

$$A = -\frac{L}{2} \sin \alpha_1 - L \sin (\alpha_1 + \alpha_2) - L \sin (\alpha_1 + \alpha_2 + \alpha_3) - L \sin (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) \quad (3.51)$$

$$B = -L \sin (\alpha_1 + \alpha_2) - L \sin (\alpha_1 + \alpha_2 + \alpha_3) - L \sin (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) \quad (3.52)$$

$$C = -L \sin (\alpha_1 + \alpha_2 + \alpha_3) - L \sin (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) \quad (3.53)$$

$$D = -L \sin (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) \quad (3.54)$$

$$E = \frac{L}{2} \cos \theta_1 + L \cos (\alpha_1 + \alpha_2) + L \cos (\alpha_1 + \alpha_2 + \alpha_3) + L \cos (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) \quad (3.55)$$

$$F = L \cos(\alpha_1 + \alpha_2) + L \cos(\alpha_1 + \alpha_2 + \alpha_3) + L \cos(\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) \quad (3.56)$$

$$G = L \cos(\alpha_1 + \alpha_2 + \alpha_3) + L \cos(\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) \quad (3.57)$$

$$H = L \cos(\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) \quad (3.58)$$

$$J(\Theta) = \begin{bmatrix} A & B & C & D \\ E & F & G & H \end{bmatrix} \quad (3.59)$$

La ecuación dada para hallar el vector de velocidad angular se encuentra en la ecuación 3.60:

$$\dot{\Theta} = J(\Theta)^{-1} \dot{X}_{deseada} \quad (3.60)$$

Donde la matriz  $J(\Theta)^{-1}$  puede ser cuadrada, caso en que genera que la matriz  $J(\Theta)$  puede ser invertida a través del método general dando como resultado la ecuación 3.62:

$$J(\Theta) = \begin{bmatrix} -\frac{L}{2} \sin(\alpha_1) - L \sin(\alpha_1 + \alpha_2) & -L \sin(\alpha_1 + \alpha_2) \\ \frac{L}{2} \cos(\alpha_1) + L \cos(\alpha_1 + \alpha_2) & L \cos(\alpha_1 + \alpha_2) \end{bmatrix} \quad (3.61)$$

$$\begin{bmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \end{bmatrix} = J(\Theta)^{-1} \begin{bmatrix} \dot{X}_{deseada} \\ \dot{Y}_{deseada} \end{bmatrix} \quad (3.62)$$

Pero al estudiar una matriz que no es cuadrada es necesario encontrar una solución pseudoinversa es denotada de la siguiente forma:

$$\dot{\Theta} = J(\Theta)^+ \dot{X}_{deseada} \quad (3.63)$$

Donde  $J(\Theta)^+$  opta la forma de  $A^T (AA^T)^{-1}$  ocasionando que la solución sea dada por la descomposición en valores singulares (SVD) dado que  $J(\Theta)^+ = U * \Sigma^+ * V^T$  encontrando la ecuación 3.64:

$$\Theta = U * \sum * V^T * \dot{X}_{deseada} \quad (3.64)$$

Pero es necesario entender que la pseudoinversa no posee una respuesta única, debido a que el resultado es uno de los posibles resultados que la diferencia de la inversa verdadera.

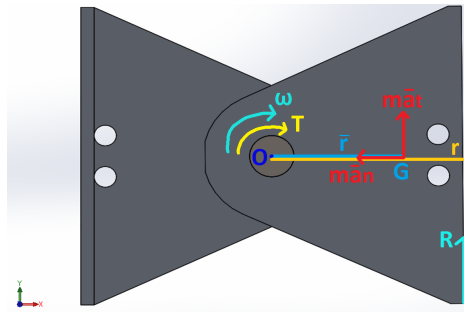
## Capítulo 4

# ESTUDIO DE MATERIAL

### 4.1. Análisis dinámico de un módulo

El análisis dinámico de un módulo se realiza con el fin de estudiar la fuerza de reacción que debe soportar la estructura con el objetivo de hacer el diseño. Empleando la (Figura 4.1), exponiendo el diagrama de cuerpo libre resaltando la velocidad angular, torques y reacciones generadas en el módulo junto a la aceleración normal ( $\bar{a}_n$ ), aceleración tangencial ( $\bar{a}_t$ ) y el parámetro de distancia del centro de gravedad a centro de apoyo ( $\bar{r}$ ).

Figura 4.1: Módulo de robot ápedo



Fuente: Elaboración propia

Con objetivo de simplificar el estudio dinámico del sistema planteado, se considera que se utilizan servomotores por más de 30 milisegundos generando una velocidad angular ( $\omega$ ) constante, por lo tanto la aceleración angular denominada normalmente como  $\alpha$  es el resultado de la derivada de la velocidad angular ocasionando que su valor sea igual a cero ( $\alpha = 0$ ) para este caso.

Usando como apoyo el libro «*Mecánica vectorial para ingenieros: Dinámica*» de Ferdinand Pierre Beer [2] en la pagina 1095, problema resuelto 17.4, sabiendo que la velocidad angular empleada es una constante  $\omega$ , se generan dos componentes desde el

centro de gravedad  $G$  del módulo que representan la aceleración normal y la aceleración tangencial mostradas en la ecuaciones 4.1 y 4.2 respectivamente, en el cual  $\bar{r}$  es el radio del centro de gravedad hasta el radio de giro y  $\dot{r}$  es la derivada del radio respecto al tiempo, pero al ser una constante el valor de la variable  $\dot{r}$  es igual a 0.

$$\bar{a}_n = \bar{r}\omega^2 + \dot{r} \quad (4.1)$$

$$\bar{a}_t = \bar{r}\alpha + \omega\dot{r} \quad (4.2)$$

Usando como fundamento el diagrama de cuerpo libre exhibido en la (*Figura 4.1*) se encuentra la suma de momentos con base al punto de apoyo « $O$ » (ecuación 4.3) donde  $r$  es la distancia de centro de giro hasta el punto donde se encuentra la fuerza, otra ecuación es la suma de fuerzas en el eje X (ecuación 4.4) donde se encuentra la reacción de las fuerzas en el eje X  $R_x$ , la masa del módulo  $m$  y la aceleración tangencial y finalmente la suma de fuerzas en el eje Y (ecuación 4.5) donde se encuentra la reacción de las fuerzas en el eje Y  $R_y$ , el peso del módulo  $p$ , la masa del módulo  $m$  y la aceleración normal.

$$\sum M_O = \bar{I}\alpha + m\bar{a}_t r \quad (4.3)$$

$$\sum F_x = R_x = m\bar{a}_t \quad (4.4)$$

$$\sum F_y = R_y - p = -m\bar{a}_n \quad (4.5)$$

Considerando las condiciones mencionadas por el servomotor las ecuaciones 4.3 y 4.4 son iguales a cero por lo tanto  $R_x = 0$ ,  $\dot{r} = 0$  y  $\alpha = 0$  ocasionando que la ecuación principal donde se encuentra la fuerza de interés para el análisis  $R_y$  sea la única que genere reacción. Con ese objetivo se reemplaza la ecuación 4.1 en la ecuación 4.5 y se despeja la fuerza de reacción en el eje y  $R_y$  sin considerar  $\dot{r}$  al no interferir en el resultado, encontrando la ecuación 4.6.

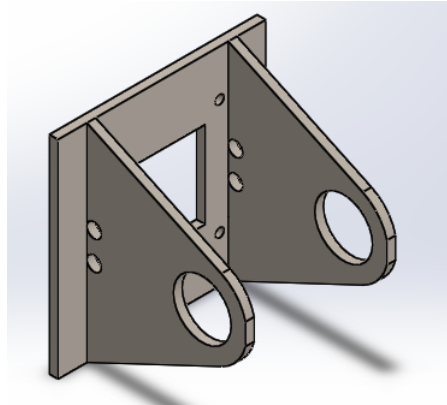
$$R_y = -m * \bar{r} * \omega^2 - \dot{r} + p \quad (4.6)$$

## 4.2. Volumen de pieza principal

Con el fin de realizar el estudio de los materiales, es necesario realizar el análisis del diseño del robot ápedo, teniendo en cuenta diferentes características como son: la estructura, los esfuerzos, las fuerzas y por supuesto, el material a utilizar. El diseño en que se basó este trabajo de grado es el «*Cube Reloaded* [12]», visto en la (*Figura 2.4*) debido a que permite el correcto movimiento sinusoidal y refleja de manera adecuada el funcionamiento de las celdas unitarias en la estructura del robot, lo que permite estudiar

de mejor manera el sistema de locomoción. Para realizar el estudio de la estructura es necesario analizar cada una de las celdas unitarias, hallando su respectivo volumen con el fin de encontrar el peso total del robot. Para lograr este objetivo es necesario realizar primero el cálculo del volumen de la pieza principal vista en la (*Figura 4.2*).

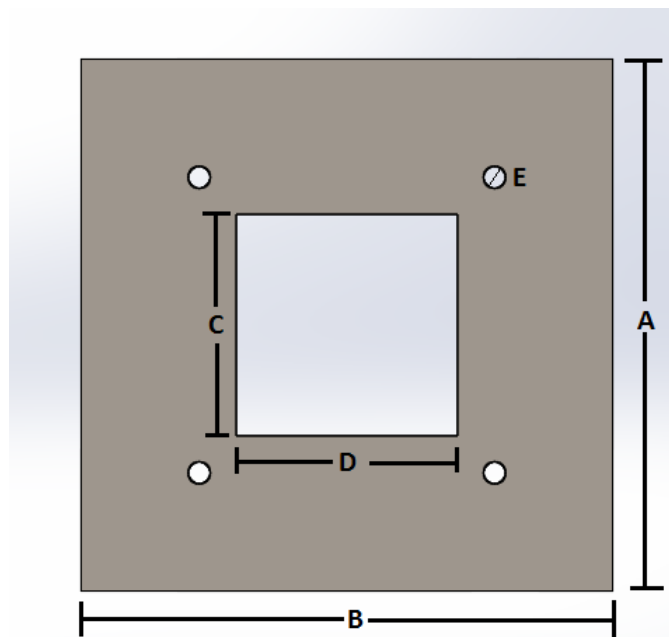
Figura 4.2: Pieza Pincipal



Fuente: Elaboración propia

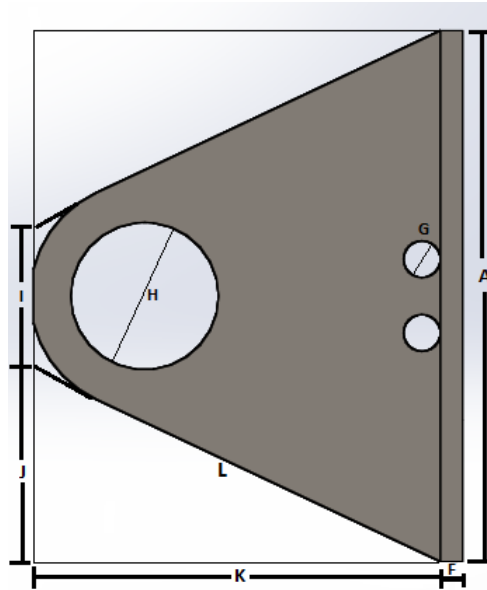
Para hallar el volumen de la pieza mencionada anteriormente, se analiza usando diferentes vistas con base al plano coordenado mostradas en las (*Figura 4.3*) y (*Figura 4.4*) vistas a continuación.

Figura 4.3: Pieza principal -vista posterior



Fuente: Elaboración propia

Figura 4.4: Pieza principal-vista lateral.



Fuente: Elaboración propia

Para realizar los cálculos del volumen se toma la vista de la (*Figura 4.3*) y debido a que es un cubo, se tiene en cuenta que el volumen de un cubo es la multiplicación del ancho por la altura por la profundidad, y que se observa en la ecuación 4.7.

$$V = A * B * F \quad (4.7)$$

Debido a que el cubo tiene varios agujeros, se procede a sacar el volumen de cada uno de estos huecos para posteriormente realizar la resta con el volumen total del cubo comprendido entre A, B, F. Como primer paso, se halla el volumen del cubo comprendido entre C, D y F que corresponde a la profundidad.

$$V_{\text{cubo pequeño}} = C * D * F \quad (4.8)$$

Posteriormente, se halla el volumen de un cilindro y se multiplica por cuatro debido a que son cuatro los cilindros que se encuentran en esta cara de la pieza. Se tiene en cuenta que el volumen del cilindro se encuentra de la siguiente manera

$$V_{\text{cilindro}} = \Pi * \text{radio}^2 * \text{altura} \quad (4.9)$$



por lo tanto

$$V_{cilindro} = \Pi * \left(\frac{E}{2}\right)^2 * F \quad (4.10)$$

Luego

$$V_{4cilindros} = V_{cilindro} * 4 \quad (4.11)$$

Para encontrar el volumen total de esta cara de la pieza se tiene que

$$V_{totalvista1} = V - V_{cubopequeño} - V_{4cilindros} \quad (4.12)$$

Como segunda parte, se toma la vista que se observa en la (*Figura 4.4*), y se calcula el valor de J de la siguiente manera:

$$J = \frac{A - I}{2} \quad (4.13)$$

y posteriormente se calcula el área del triángulo comprendido entre K, J, y L como se puede ver a continuación:

$$A_{triángulo} = \frac{K * J}{2} \quad (4.14)$$

A continuación se multiplica el área del triángulo hallado anteriormente por F que es la profundidad, y luego esta multiplicación se vuelve a multiplicar por dos ya que existen dos triángulos iguales.

$$V_{totaltriángulos} = (A_{triángulo} * 3) * 2 \quad (4.15)$$

Seguidamente, se calcula el volumen de los cilindros.

$$V_{cilindrospequeños} = \left( \Pi * \left(\frac{G}{2}\right)^2 * F \right) * 2 \quad (4.16)$$

$$V_{cilindrogrande} = \left( \Pi * \left(\frac{H}{2}\right)^2 * F \right) \quad (4.17)$$

Por último, se realiza el cálculo del volumen total y se hace la resta del volumen total de la pieza menos el volumen de los cilindros

$$V_2 = A * K * F \quad (4.18)$$

$$V_{totalvista2} = V_2 - V_{totaltriángulos} - V_{cilindrospequeños} - V_{cilindrogrande} \quad (4.19)$$

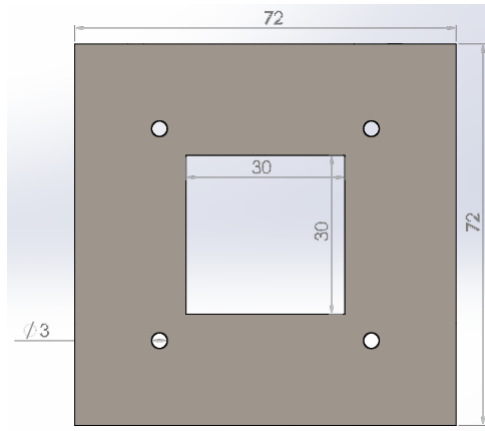
Y considerando que se replica en su segunda pestaña el volumen encontrado de debe multiplicar por dos para así finalmente, realizar la sumatoria de volúmenes de la vistas para hallar el volumen total de la pieza uno.

$$V_{totalpieza1}=V_{totalvista1} + V_{totalvista2} * 2 \quad (4.20)$$

#### 4.2.1. Implementación del volumen en la pieza

Para hacer la implementación del volumen de la pieza usada como objeto de estudio es necesario sacar las diferentes medidas que se van a utilizar con ayuda del programa *Solidworks*, como se observa a continuación en la (Figura 4.5):

Figura 4.5: Pieza principal acotada en milímetros



Fuente: Elaboración propia

Luego, se realiza el procedimiento explicado en la sección 4.2, tomando primero la cara vista en la (Figura 4.5), debido a que es un cubo, se tiene en cuenta que el volumen de un cubo es la multiplicación del ancho por la altura por la profundidad, y tomando las medidas en milímetros, se tiene la siguiente ecuación:

$$V = 72 * 72 * 3 = 15552mm^3 \quad (4.21)$$

Debido a que el cubo tiene varios agujeros, se procede a sacar el volumen de cada uno de estos huecos para posteriormente realizar la resta con el volumen total del cubo. Como primer paso, se halla el volumen del cubo pequeño:

$$V_{cubopequeño} = 30 * 30 * 3 = 2700mm^3 \quad (4.22)$$

Después, se halla el volumen de un cilindro y se multiplica por cuatro debido a que son cuatro los cilindros que se encuentran en esta cara de la pieza. Se tiene en cuenta que el volumen del cilindro se encuentra de la siguiente manera:

$$V_{cilindro} = \Pi * 1,5^2 * 3 = 21,20575041mm^3 \quad (4.23)$$

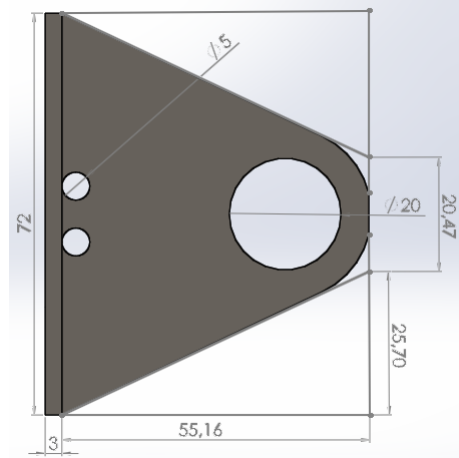
$$V_{4cilindros} = 21,20575041 * 4 = 84,82300165mm^3 \quad (4.24)$$

Para encontrar el volumen total de esta cara de la pieza se tiene que:

$$V_{totalvista1} = 15552 - 2700 - 84,82300165 = 12767,177mm^3 \quad (4.25)$$

Como segunda parte, se toma la vista que se observa en la (Figura 4.6) vista a continuación:

Figura 4.6: Pieza principal vista lateral acotada en milímetros



Fuente: Elaboración propia

Y con ayuda del programa *Solidworks* se sacan las diferentes medidas que se utilizan para calcular el volumen de esta cara de la pieza. Como primer paso para hallar el volumen de esta cara de la pieza se halla el área del triángulo, teniendo en cuenta que las medidas están en milímetros, de la siguiente manera:

$$A_{triángulo} = \frac{55,16 * 25,70}{2} = 708,806mm^2 \quad (4.26)$$

A continuación se multiplica el área del triángulo hallado anteriormente por 3 mm que es la profundidad, y luego esta multiplicación se vuelve a multiplicar por dos ya que existen dos triángulos iguales.

$$V_{totaltriángulos} = (708,806 * 3) * 2 = 4252,836mm^3 \quad (4.27)$$

Seguidamente, se calcula el volumen de los cilindros.

$$V_{cilindrospequeños} = \left( \Pi * \left( \frac{5}{2} \right)^2 * 3 \right) * 2 = 117,8097245mm^3 \quad (4.28)$$

$$V_{cilindrogrande} = \Pi * \left( \frac{20}{2} \right)^2 * 3 = 942,4777961mm^3 \quad (4.29)$$

Por último, se realiza el cálculo del volumen total y se hace la resta del volumen total de la pieza menos el volumen de los cilindros.

$$V_2 = 72 * 55,16 * 3 = 11914,56mm^3 \quad (4.30)$$

$$V_{totalvista2} = 11914,56 - 4252,836 - 117,8097245 - 942,4777961 = 6601,436479mm^3 \quad (4.31)$$

Y considerando que se replica en su segunda pestaña el volumen encontrado de debe multiplicar por dos para así finalmente, realizar la sumatoria de volúmenes de la vistas para hallar el volumen total de la pieza uno.

$$V_{totalpieza1} = 12767,177 + 6601,436479 * 2 = 25970,04996mm^3 \quad (4.32)$$

Pasando el volumen total de la pieza uno en metros se obtiene que:

$$V_{totalpieza1} = 25970,04996 * \frac{1}{1000^3} = 0,025970049 * 10^{-3}m^3 \quad (4.33)$$

### 4.3. Parámetros de material

Para realizar el estudio de material se consideran las características del motor seleccionado (servomotor SG-5010 de Tower pro) expuestos en el cuadro 4.1, usado como base para realizar los cálculos correspondientes para realizar el adecuado análisis.

Cuadro 4.1: Características del servomotor SG-5010 de Tower pro

Característica	Valor
Torque a 4.8V	5.50Kg-cm
Torque a 6.0V	6.50Kg-cm
Velocidad a 4.8V	0,19sec/60°
Velocidad a 6.0V	0.15 sec/60°
Peso( $m_s$ )	47g

Fuente: Copyright[5]

Considerando los materiales empleados para la construcción de los robots ápodos (Acero 10-20, Aluminio, ABS y fibra de carbono) y el volumen dado por la ecuación 4.20 se halla la masa de la pieza principal que forma la estructura principal del robot objeto de estudio, a través de la ecuación 4.34.

$$m = \delta_{densidad} * V_{Volumen} \quad (4.34)$$

Al multiplicar la masa de la pieza principal por el número de módulos que componen la estructura del robot ápodo ( $M = 9$ ) en este caso, se obtiene la masa de la estructura completa del robot ( $m_r$ ). Teniendo en cuenta que cada módulo requiere un servomotor que pesa  $m_s$ , y considerando elementos adicionales ( $m_o$ ) como la fuente de energía, rodamiento, etc. La ecuación resultante para la masa es la ecuación 4.35.

$$m_r = m * M * 2 + m_s * M + m_o \quad (4.35)$$

Al multiplicar la ecuación 4.35 correspondiente a la masa total por la fuerza de gravedad ( $9,81m/s^2$ ) se obtiene el peso total del robot ( $P_t$ ), dando como resultado la ecuación 4.36.

$$P_t = m_r * 9,81 \quad (4.36)$$

## 4.4. Materiales objeto de estudio

Los materiales usados como objeto de estudio usados en este trabajo son acero 10-20, Aluminio, ABS y fibra de carbono al ser los comúnmente utilizados para la construcción de los robots ápodos.

### 4.4.1. Acero 1020

El acero 1020 es un acero con mayor fortaleza a comparación de otros aceros como el 1018 y posee las cualidades de baja tenacidad y baja resistencia mecánica por lo que es adecuado para fabricar elementos de maquinaria  $7870Kg/m^3$ . Con el objetivo de encontrar la masa de la pieza con base a ese materia se usa la ecuación 4.34 obteniendo como resultado la ecuación 4.37.

$$m = 7870 * 0,025970049 * 10^{-3} = 0,2043842856Kg \quad (4.37)$$

Contemplando el peso del servomotor utilizado en el caso de estudio es de 0.047Kg según la tabla 4.1 y los nueve módulos que conforman el cuerpo del robot en este caso ( $M = 9$ ) y reemplazando en la ecuación 4.35 se encuentra la masa total del robot dando como resultado la ecuación 4.38.

$$m_r = 0,2043842856 * 9 * 2 + 0,047 * 9 + 0,247 = 4,348917141Kg \quad (4.38)$$

Es requerido encontrar el peso total de la ecuación 4.35 encontrando la ecuación 4.39 que expone el resultado de la fórmula.

$$P_t = 4,348917141 * 9,81 = 42,66287716N \quad (4.39)$$

Contemplando los resultados de las ecuaciones anteriormente utilizadas, la velocidad máxima que alcanza el servomotor que es de  $0,19sec/60^\circ$  que es equivalente a  $0,57sec/\Pi$  y la distancia desde el centro de gravedad localizado a través del simulador hasta el centro del punto de apoyo que corresponde a  $44,6 * 10^{-3}m$ . Se usa la ecuación 4.6 que describe la reacción en el eje y del módulo, alcanzando los resultados presentados en la ecuación 4.40.

$$R_y = -4,348917141 * 44,6 * 10^{-3} * 0,57^2 + 42,66287716 = 42,599859N \quad (4.40)$$

Encontrando la totalidad de fuerzas que interactúan en un sólo módulo considerando que el esfuerzo máximo que debe ejercer soporta el peso total del robot.

#### 4.4.2. Aluminio

El aluminio es un material metálico con densidad de  $2698,4Kg/m^3$  por lo que aplicando la ecuación 4.34 se puede obtener la masa de la pieza principal en la ecuación 4.41.

$$m = 2698,4 * 0,025970049 * 10^{-3} = 0,07009067016Kg \quad (4.41)$$

El peso del servomotor usado en el caso de estudio es de  $0.047Kg$  según la tabla 4.1 y el número de módulos que conforman el cuerpo del robot ( $M = 9$ ) y reemplazando en la ecuación 4.35 encontrando la masa total del robot dando como resultado la ecuación 4.42.

$$m_r = 0,07009067016 * 9 * 2 + 0,047 * 9 + 0,247 = 1,931632063Kg \quad (4.42)$$

Con objetivo de encontrar el peso total del robot se utiliza la ecuación 4.36 hallando el resultado en la ecuación 4.43.

$$P_t = 1,931632063 * 9,81 = 18,94931054N \quad (4.43)$$

Considerando los resultados de las ecuaciones anteriormente implementadas, la velocidad del servomotor máxima de  $0,19sec/60^\circ$  es decir  $0,57sec/\Pi$  y la distancia desde el centro de gravedad localizado a través del simulador hasta el centro del punto de apoyo que es de  $44,6 * 10^{-3}m$ . Utilizando la ecuación 4.6 que describe la reacción en el eje y del módulo, obteniendo los resultados presentados en la ecuación 4.44.

$$R_y = -1,931632063 * 44,6 * 10^{-3} * 0,57^2 + 18,94931054 = 18,92132015N \quad (4.44)$$

Encontrando la totalidad de las fuerzas que interactuarán en el módulo en caso extremo en que el módulo tendría que soportar el peso de toda la estructura hecha de aluminio principalmente.

#### 4.4.3. ABS (acrilonitrilo butadieno estireno)

El ABS también conocido como acrilonitrilo butadieno estireno, es conocido por ser un termoplástico duro resistente al impacto con una fuerza al impacto de  $10-20Kj/m^2$  y densidad de  $1050Kg/m^3$ . Buscando la masa de la pieza objeto estudiada se utiliza la

ecuación 4.34 usando los valores correspondientes al material seleccionado dando como resultado la ecuación 4.45.

$$m = 1050 * 0,025970049 * 10^{-3} = 0,02726855145Kg \quad (4.45)$$

Teniendo en cuenta el peso del servomotor utilizado en este caso de estudio de 0.047Kg según la tabla 4.1 y el número de módulos que conforman la estructura del robot ( $M = 9$ ) y reemplazando en la ecuación 4.35 encontrando la masa total del robot dando como resultado la ecuación 4.46.

$$m_r = 0,02726855145 * 9 * 2 + 0,047 * 9 + 0,247 = 1,160833926Kg \quad (4.46)$$

Encontrando el peso total del robot a través de la ecuación 4.36 se puede encontrar los resultados mostrados en la ecuación 4.47.

$$P_t = 1,160833926 * 9,81 = 11,38778082N \quad (4.47)$$

Con base a los resultados de las ecuaciones empleadas anteriormente, la velocidad del servomotor máxima de  $0,19sec/60^\circ$  equivalente a  $0,57sec/\Pi$  y la distancia desde el centro de gravedad hallado a través del simulador hasta el centro del punto de apoyo correspondiente a  $44,6 * 10^{-3}m$ , se reemplaza la totalidad de las variables encontradas en la ecuación 4.6 que describe la reacción en el eje y del módulo. Dando los resultados presentados en la ecuación 4.48.

$$R_y = -1,160833926 * 44,6 * 10^{-3} * 0,57^2 + 11,38778082 = 11,37095971N \quad (4.48)$$

Despejando las fuerzas que ejercen sobre el módulo considerando el esfuerzo máximo equivalente al peso completo del robot, debido a que un módulo debe soportar el impacto de la totalidad del cuerpo.

#### 4.4.4. Fibra de carbono

La fibra de carbono es un polímero no metálico con densidad de  $1,750Kg/m^3$ , posee cualidades como baja conductividad térmica y resistencia a la conductividad eléctrica. Sus propiedades físicas son similares a las presentadas en el acero pero con la diferencia de ser ligero como la madera o plástico sin perder resistencia. La masa requerida para implementar el diseño planteado se utiliza en la ecuación 4.34, obteniendo como resultado la ecuación 4.49 al reemplazar el volumen total encontrado anteriormente y la densidad del material.

$$m = 1,750 * 0,025970049 * 10^{-3} = 4,544758575 * 10^{-5}Kg \quad (4.49)$$

Agregando el peso del servomotor utilizado de 0.047Kg según la tabla 4.1 y teniendo en cuenta el número de módulos utilizado en el robot ( $M = 9$ ), es reemplazado en

la ecuación 4.35 encontrando la masa total del robot obteniendo como resultado la ecuación.

$$m_r = 4,544758575 * 10^{-5} * 9 * 2 + 0,047 * 9 + 0,247 = 0,6708180565Kg \quad (4.50)$$

Buscando el peso total del robot con base al material estudiado se utiliza la ecuación 4.36, dando como resultado la ecuación 4.51.

$$P_t = 0,6708180565 * 9,81 = 6,580725135N \quad (4.51)$$

Utilizando los resultados de las ecuaciones anteriormente encontradas junto con la velocidad del servomotor máxima de  $0,19sec/60^\circ$  equivalente a  $0,57sec/\Pi$  y la distancia desde el centro de gravedad hallado a través del simulador hasta el centro del punto de apoyo correspondiente a  $44,6 * 10^{-3}m$ . Teniendo el valor de las variables y considerando la ecuación 4.6 al representar la reacción resultante en el eje y del módulo, obteniendo la ecuación de resultado 4.52.

$$R_y = -0,6708180565 * 44,6 * 10^{-3} * 0,57^2 + 6,580725135 = 6,571004619N \quad (4.52)$$

Con la ecuación 4.52 se termina de encontrar las fuerzas que ejercen en el módulo considerando que la fuerza máxima es equivalente al peso completo del robot por lo que debe soportar el impacto del generado por el cuerpo completo del robot.

## 4.5. Simulación de esfuerzos

La simulación de esfuerzos es realizada usando la herramienta de diseño *Solidworks* con el objetivo de analizar el comportamiento de las piezas del robot, considerando el valor de las fuerzas encontradas anteriormente como parámetros primordiales frente a la simulación de esfuerzos al ser las principales fuerzas que interactúan con el robot en caso extremo. Gracias a que se está considerando el caso de un módulo en circunstancias extremas, es decir que sea capaz de soportar el esfuerzo máximo proporcionado por el peso completo del robot se logra localizar y definir los puntos de apoyo y magnitud de la fuerza entrante siendo evaluado en estado estático con un factor de seguridad de 1.

El módulo principal del robot objeto de estudio fue sometido a las fuerzas principales que interactúan con él, compuestas por la fuerza que soportaría con base a la fuerza de la estructura completa que tendría que soportar y la fuerza del torque que ejerce el servomotor a la pieza.

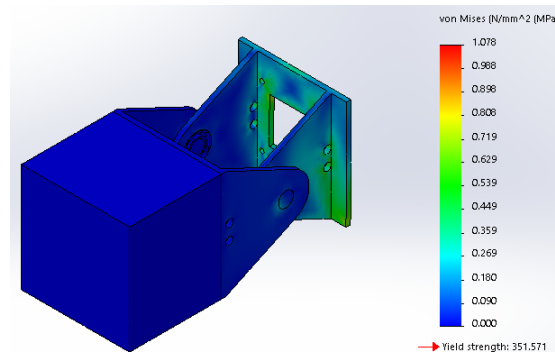
### 4.5.1. Acero 1020

En la (*Figura 4.7*) se refleja como el punto de impacto de la pieza principal donde soporta una presión de  $0,719N/mm^2$  (*MPa*) y se distribuye entorno a la pieza, en la cual



se visualizan esfuerzos aproximados a  $0,539\text{N/mm}^2$  ( $\text{MPa}$ ) junto con un esfuerzo en el punto de apoyo entre las piezas de  $0,359\text{N/mm}^2$  ( $\text{MPa}$ ). El límite elástico que posee la estructura es de  $351,571\text{N/mm}^2$  ( $\text{MPa}$ ) por lo que no sufre deformación permanente al impacto de la fuerza implementadas en la simulación.

Figura 4.7: Acero 1020-Esfuerzos

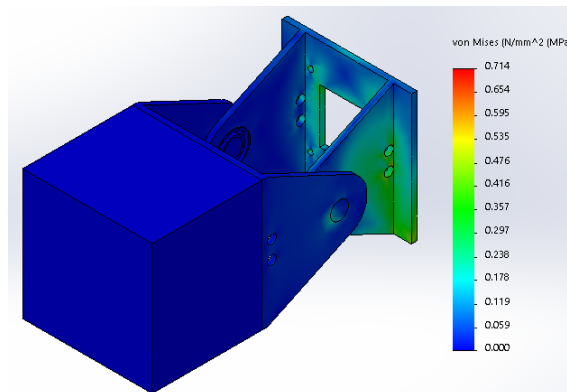


Fuente: Elaboración propia

#### 4.5.2. Aluminio

En la (*Figura 4.8*) se muestran los esfuerzos resultantes del módulo principal del robot, en el cual se halla una presión de  $0,535\text{N/mm}^2$  ( $\text{MPa}$ ) en el punto de impacto principal del módulo y múltiples esfuerzos distribuidos variando entre  $0,119\text{N/mm}^2$  ( $\text{MPa}$ ) hasta  $0,476\text{N/mm}^2$  ( $\text{MPa}$ ) de forma dispersa con origen en el punto del impacto junto con un esfuerzo de aproximadamente  $0,238\text{N/mm}^2$  ( $\text{MPa}$ ) en el punto transmisión de fuerza con el servomotor.

Figura 4.8: Aluminio-Esfuerzos

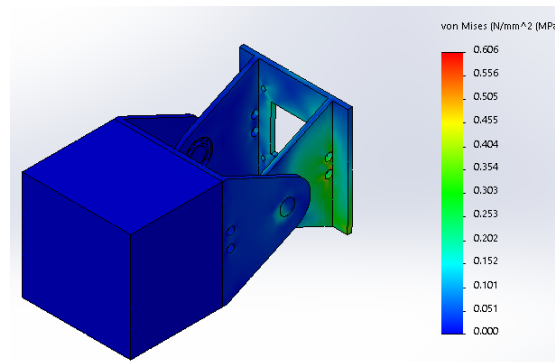


Fuente: Elaboración propia

### 4.5.3. ABS

Los esfuerzos en el módulo con base al material acrilonitrilo butadieno estireno se exhiben en la (Figura 4.9) donde una presión de  $0,404N/mm^2$  (MPa) es localizada en el punto de impacto y dispersándose en torno a la pieza principal adquiriendo valores desde  $0,152N/mm^2$  (MPa) hasta  $0,354N/mm^2$  (MPa). En torno a los puntos de apoyo del servomotor se concentran los esfuerzos de mayor valor a alcanzar en medida mínima la presión de  $0,606N/mm^2$  (MPa), a diferencia del punto de apoyo para la transmisión de movimiento a la pieza.

Figura 4.9: ABS-Esfuerzos



Fuente: Elaboración propia

## 4.6. Simulación de deformación

Al realizar la simulación de elementos finitos, también se puede obtener la deformación que sufren las piezas con base a los esfuerzos realizados resultantes en el módulo principal objeto del estudio realizado. Con el fin de verificar el comportamiento adecuado de la estructura en base a la fuerza extrema, a las que sería sometido en un caso extremo donde soportaría el peso completo de la estructura junto a la fuerza de torque que ejercería el servomotor el motor y definiendo el cumplimiento de los parámetros adecuados para la futura implementación.

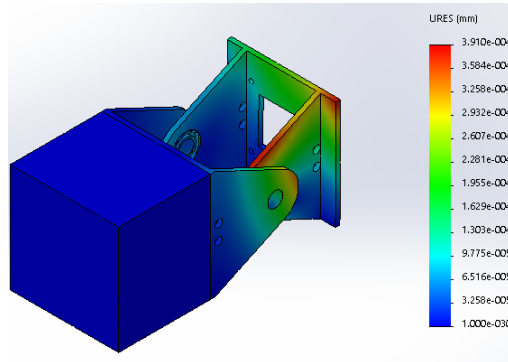
Para lograr los objetivos se evaluaron los resultados de las deformaciones en la pieza con base al material seleccionado exhibiendo los puntos críticos donde se refleja la máxima deformación.

### 4.6.1. Acero 1020

En la (Figura 4.10) se observan múltiples deformaciones en base a las fuerzas que interactúan con el módulo, donde la mayor deformación ocurre en la esquina superior derecha y en la arista de esa cara de la pieza, debido a que es la secciones de la pieza que

genera mayor impacto al estar conectada directamente a la posición del servomotor y al punto de impacto. Gracias a eso se crea una deformación máxima de  $3,910 * 10^{-0,04}mm$  y dispersándose alrededor de la pieza múltiples deformaciones entre  $3,258 * 10^{-0,05}mm$  hasta  $3,584 * 10^{-0,04}mm$ . Las deformaciones fueron generadas también en el punto de transmisión de movimiento por el servomotor con valores similares a los ya mencionados de  $3,258 * 10^{-0,05}mm$  y  $3,584 * 10^{-0,04}mm$ .

Figura 4.10: Acero 1020-Deformación.

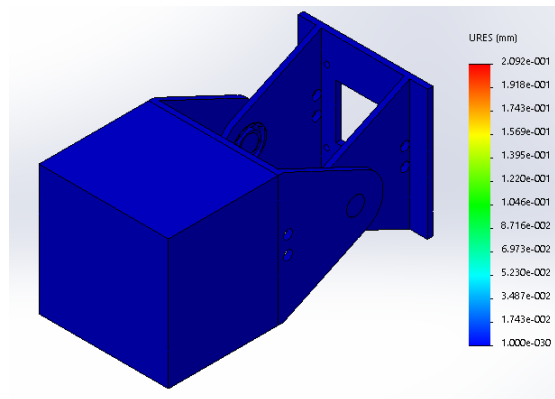


Fuente: Elaboración propia

#### 4.6.2. Aluminio

La (Figura 4.11) se observan los resultados de la deformación causada por los esfuerzos del módulo principal del robot objeto de estudio. La deformación resultante fue prácticamente nula al presentar deformación constante en toda la estructura de  $1 * 10^{-0,3}mm$  y no mostrar otro tipo de cambio adicional con base al módulo, ocasionando que sea un material apropiado para la fabricación del robot estudiado.

Figura 4.11: Aluminio- Deformación

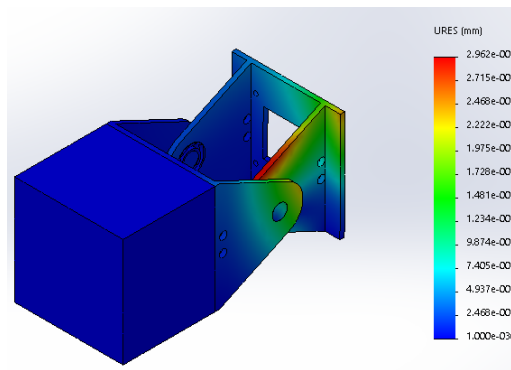


Fuente: Elaboración propia

### 4.6.3. ABS

Empleando el material ABS mostrado en la (*Figura 4.12*) se expusieron deformaciones concentradas en la arista del lado derecho de la pieza principal del módulo donde es aplicada la fuerza de locomoción del robot proporcionado por el servomotor y es transmitida la fuerza ejercida por el impacto estimulando una deformación máxima de  $2,962 * 10^{-0,02} mm$  junto con deformaciones que se originan en ese punto encontrados entre  $2,468 * 10^{-0,03} mm$  y  $2,468 * 10^{-0,02} mm$ . También se generaron deformaciones similares en el punto de apoyo del motor donde transmite el movimiento a la pieza principal, ocasionando que las deformaciones se concentraran en esa sección del módulo.

Figura 4.12: ABS- Deformación



Fuente: Elaboración propia

## 4.7. Resultados de materiales

Los materiales estudiados el capítulo mostraron diferentes cualidades y propiedades, tanto en sus reacciones como en su comportamiento frente a las diferentes fuerzas, exhibiendo los resultados en el cuadro 4.2, enfocado en las reacciones en el eje junto con los esfuerzos resultantes y las deformaciones causadas por los esfuerzos, al ser los principales factores a tener en cuenta para el análisis de los materiales.

Cuadro 4.2: Resultados de materiales

Parámetros	Reacción en el eje Y ( $N$ )	Esfuerzo mínimo ( $N/mm^2$ ) ( $MPa$ )	Esfuerzo máximo ( $N/mm^2$ ) ( $MPa$ )	Deformación mínima ( $mm$ )	Deformación máxima ( $mm$ )
Acero-1020	42.599859	0,359	0,719	$3,258 * 10^{-0,05}$	$3,910 * 10^{-0,04}$
Aluminio	18.92132015	0,119	0,535	$1 * 10^{-0,3}$	$1 * 10^{-0,3}$
ABS	11.37095971	0,152	0,606	$2,468 * 10^{-0,03}$	$2,962 * 10^{-0,02}$
Fibra de carbono	6.571004619	-	-	-	-

Fuente: Elaboración propia.

En el cuadro 4.2 se observan los diferentes resultados encontrados a través del análisis de elementos finitos, donde se obtuvieron diferentes resultados de las deformaciones y esfuerzos que sufre un módulo considerando las posibles fuerzas en que se verán afectadas. Además, de las fuerzas es posible comparar los diferentes resultados donde se destaca el aluminio al tener una deformación constante de  $1 * 10^{-0,3}$  que posee los esfuerzos más bajos, seguido por el ABS que también muestra una respuesta adecuada con diferencias mínimas frente a la deformación.

# Capítulo 5

## RESULTADOS

### 5.1. Modelo cinemático implementado

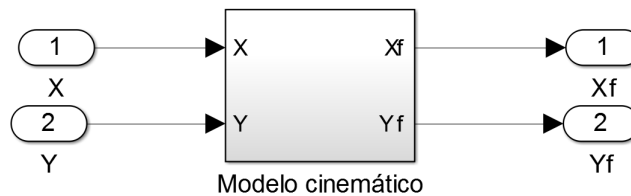
Los modelos matemáticos vistos anteriormente permiten analizar el comportamiento del robot ápedo, utilizando modelos cinemáticos comprendidos por las cinemáticas directa e inversa como se muestra en la (*Figura 5.1*), donde tiene como datos de entrada las coordenadas  $(X, Y)$  deseadas y son obtenidas las coordenadas  $(X, Y)$  resultantes de la cinemática. Para evaluar la concordancia del modelo a implementar se utilizó el modelo resultante de la cinemática directa expuesto en las ecuaciones 5.1 y 5.2 donde se considera una estructura compuesta por 2 módulos y el resultado del modelo de la cinemática inversa comprendido por las ecuaciones 3.23 y 3.44 del capítulo anterior llamado «*MODELO MATEMÁTICO*».

$$X = L \cos(\alpha_1) + L \cos(\alpha_1 + \alpha_2) \quad (5.1)$$

$$Y = L \sin(\alpha_1) + L \sin(\alpha_1 + \alpha_2) \quad (5.2)$$

Implementando la ecuaciones 5.1 y 5.2 pertenecientes a la cinemática directa y las ecuaciones 3.23 y 3.44, se obtiene el diagrama de bloques utilizando *Matlab®/Simulink* como se muestra en la (*Figura 5.1*) donde las coordenadas de entrada son dadas por el ángulo de serpenteo.

Figura 5.1: Cinemática *Matlab®/Simulink*



Fuente: Elaboración propia

## 5.2. Sistema lazo abierto

El sistema en lazo abierto expuesto en la (Figura 5.2), muestra la trayectoria fundamentada en el movimiento de serpenteo comprendido por las ecuaciones 3.5 y 3.6 (vistas en el capítulo 3 «MODELO MATEMÁTICO»). Los parámetros contemplados por la trayectoria consideran el número de ondulaciones ( $k$ ), el desfase de la trayectoria ( $\phi$ ), el ángulo de serpenteo, el número de módulos ( $M$ ), la longitud de cada módulo ( $L$ ) y son evaluados entre los valores entre 0 y la longitud total  $l$  que es el resultado del producto entre  $L$  y  $M$ , simbolizado como  $S$ .

Figura 5.2: Sistema lazo abierto

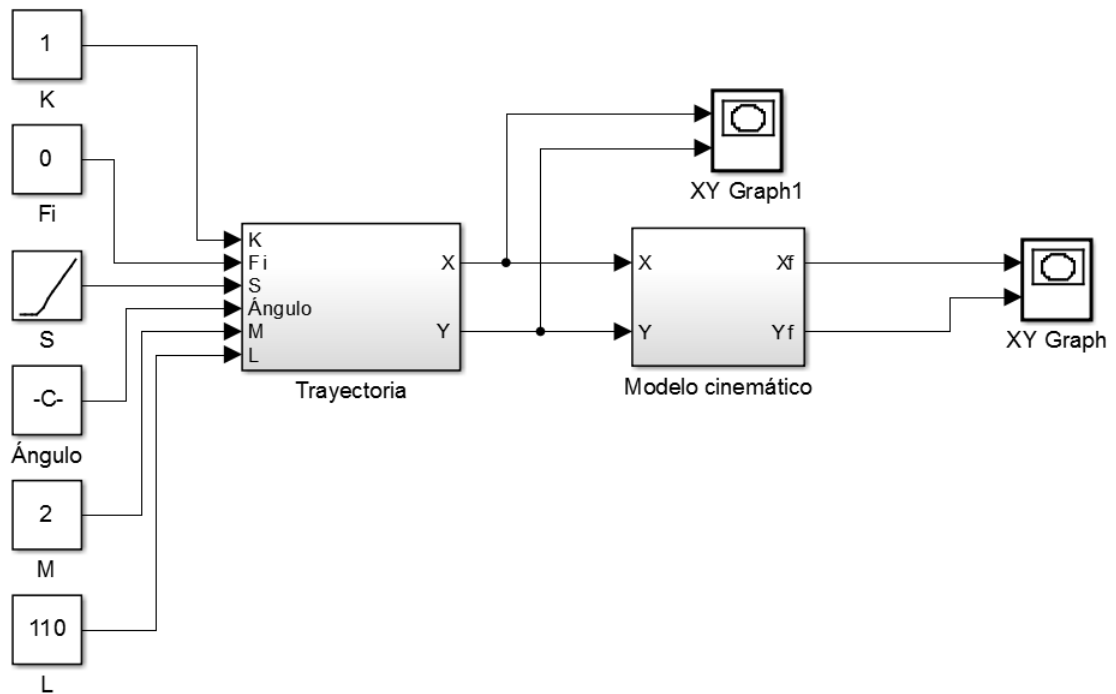


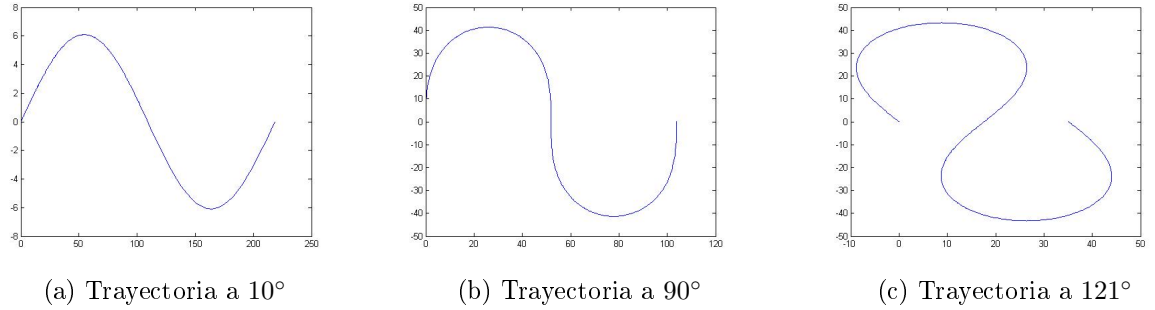
Figura: Elaboración propia

Según las ecuaciones para hallar el ángulo de serpenteo mínimo (Ecuación 3.7) y el ángulo de serpenteo máximo (Ecuación 3.8) se obtienen los ángulos de  $4,9^\circ$  y de  $30,5^\circ$  respectivamente.

### 5.2.1. Trayectoria

Realizando el estudio de la trayectoria perteneciente al movimiento de serpenteo son obtenidas las gráficas vistas en la (Figura 5.3), donde son evaluadas las diferentes formas adquiridas por la variación en el ángulo de serpenteo considerado principalmente los ángulos de  $10^\circ$ ,  $90^\circ$  y  $121^\circ$ .

Figura 5.3: Trayectorias



Fuente: Elaboración Propia

Los resultados obtenidos en la (Figura 5.3) reflejan el cambio de amplitud ( $A$ ) resultante de la relación existente entre el ángulo de serpenteo ( $\alpha$ ), el número de ondulaciones ( $k$ ), el número de módulos ( $M$ ) y la longitud de cada módulo ( $L$ ) mostrada en la ecuación 5.3, considerando las cualidades físicas planteadas para el robot ápedo.

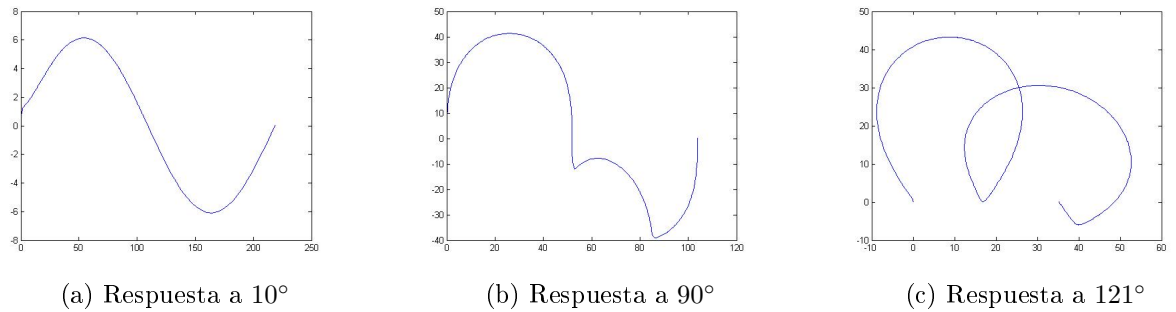
$$A = \frac{2\pi k}{M * L} \alpha \quad (5.3)$$

### 5.2.2. Respuesta del modelo cinemático

La respuesta del modelo cinemático utilizado (visto en la Figura 5.4) refleja el margen de error generado por el sistema implementado, el cual considera un error acorde al ángulo de serpenteo usado, por lo que en la gráfica correspondiente a 10° en el ángulo de serpenteo maneja un error del 9,6634% exponiendo una respuesta con un error bajo. Pero al aumentar el ángulo de serpenteo abruptamente a 90° se destaca un error del 173,9862% causada por una sección de la trayectoria, el cual aumenta aún más en la última gráfica al corresponder al ángulo de saturación máximo de 121° correspondiente a un punto crítico y genera un margen de error del 830,2372% en una sección de la gráfica exponiendo un comportamiento inadecuado del sistema por lo que es necesario utilizar un sistema de control, el cual al ser implementado permitiría al sistema movilizarse según la trayectoria deseada.



Figura 5.4: Respuesta modelo cinemático

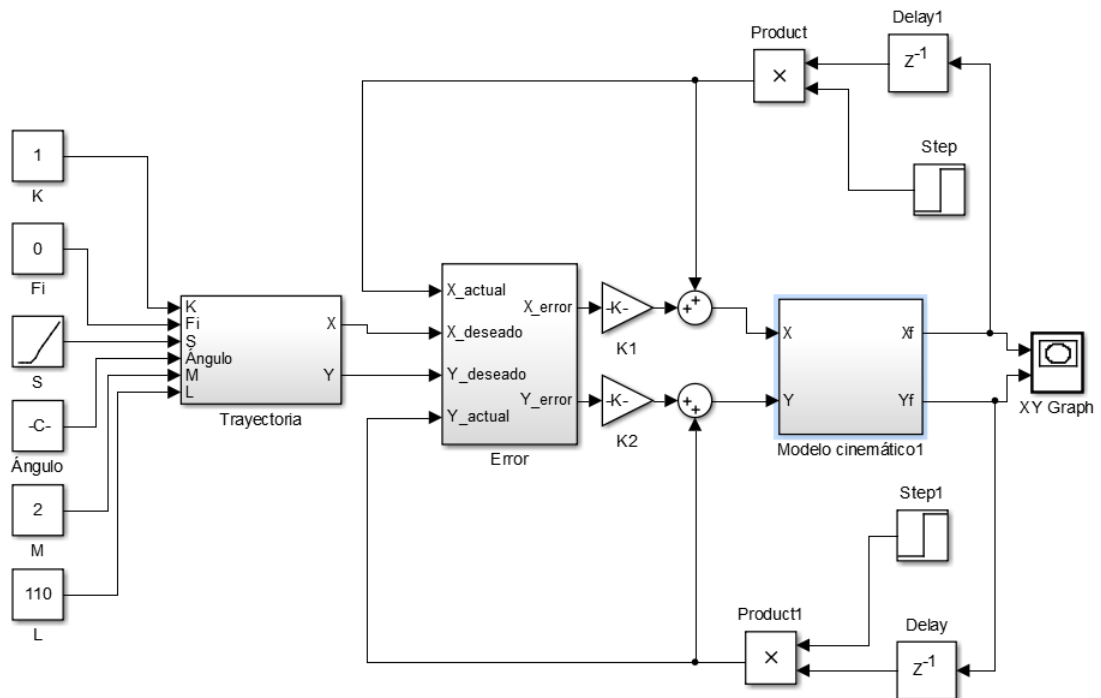


Fuente: Elaboración Propia

### 5.3. Diseño de controlador proporcional

Buscando mejorar la respuesta del sistema se utiliza un controlador proporcional con el objetivo de disminuir el tiempo de respuesta del modelo cinemático implementado. Con el fin de analizar el comportamiento del sistema frente a los cambios de las ganancias  $K_1$  y  $K_2$  mostradas en el esquema que se observa en la (Figura 5.5), el cual expone la corrección de posición a través de la retro-alimentación del sistema.

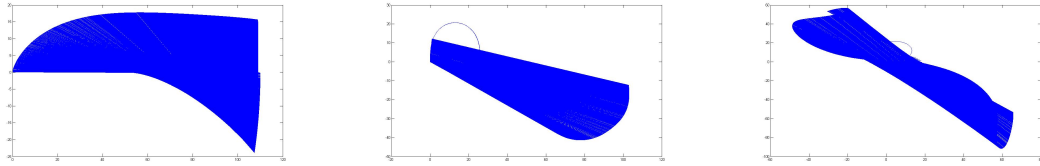
Figura 5.5: Controlador proporcional



Fuente: Elaboración propia

En la (Figura 5.6) son expuestos los resultados del controlador usando los valores de ganancia para K1 de 2 y K2 de 2. La trayectoria resultante como se observa en la (Figura 5.6) es una trayectoria no deseada que proporciona un descontrol del sistema para los distintos grados de serpenteo estudiados.

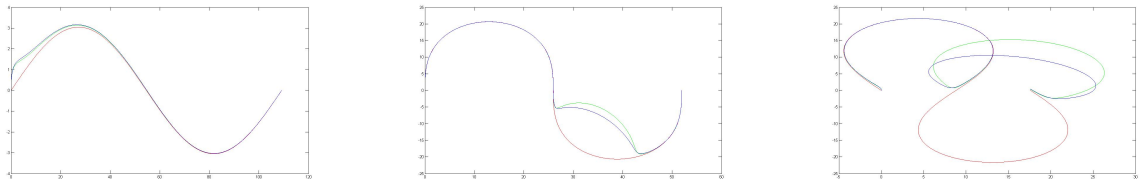
Figura 5.6: Controlador proporcional con K1=2 y K2=2



(a) Resultado con ángulo de 10° (b) Resultado con ángulo de 90° (c) Resultado con ángulo de 121°  
Fuente: Elaboración propia

La (Figura 5.7) tiene el valor de 0.8 en las ganancias de K1 y K2 con el objetivo de disminuir la magnitud del error. En las trayectorias resultantes de color rojo se observa disminución del error, la trayectoria deseada de color azul desarrollada con base al movimiento de serpenteo junto la respuesta natural del sistema de color verde.

Figura 5.7: Controlador proporcional con K1 = 0.8 y K2=0.8



(a) Resultado con ángulo de 10° (b) Resultado con ángulo de 90° (c) Resultado con ángulo de 121°  
Fuente: Elaboración propia

Al observar la variación de los resultados del sistema entorno a la variación de las ganancia obteniendo el cuadro 5.1, donde es evaluado el error del sistema considerando el error en X denotado como  $E_x$ , el error en Y denotado como  $E_y$  y el error total que esta dado por la distancia euclidiana denotada como  $E$ . Para obtener la distancia euclidiana es usada la ecuación 5.4 donde considera el error tanto en el eje x como en el eje y.

$$E = \sqrt{E_x^2 + E_y^2} \quad (5.4)$$

Cuadro 5.1: Variación de constantes del controlador proporcional

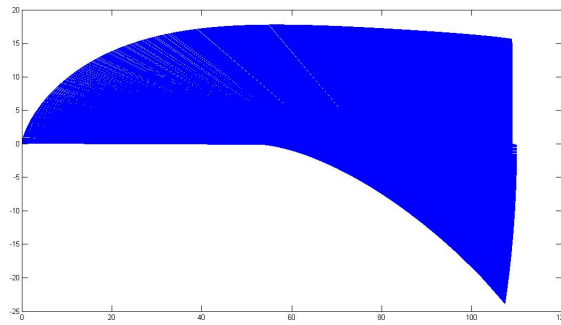
K1	K2	Ex (%)	Ey (%)	E (%)
2	2	2.1296	4.8603	5.3063
2	1.5	2.0201	7.2309	7.5077
2	1.1	1.9672	9.6659	9.8640
2	0.8	2.0247	12.9731	13.1301
2	0.2	1.5241	34.9816	35.0147
1.5	2	2.0996	7.0325	7.3392
1.5	1.5	2.2148	7.9471	8.2499
1.5	1.1	2.2303	9.7339	9.9861
1.5	0.8	2.2146	12.2307	12.4295
1.5	0.2	2.0308	34.9041	34.9631
1.1	2	2.8873	6.9005	7.4802
1.1	1.5	3.0248	7.8464	8.4092
1.1	1.1	3.0407	9.6376	10.1058
1.1	0.8	3.0174	12.1340	12.5035
1.1	0.2	2.7669	34.7926	34.9024
0.8	2	4.0140	6.7073	7.8166
0.8	1.5	4.1671	7.7046	8.7593
0.8	1.1	4.1795	9.5036	10.3820
0.8	0.8	4.1442	11.9999	12.6953
0.8	0.2	3.8002	34.6379	34.8457
0.2	2	16.7566	5.0680	17.5062
0.2	1.5	16.6940	6.3819	17.8722
0.2	1.1	16.5485	8.2318	18.4828
0.2	0.8	16.3446	10.7082	19.5399
0.2	0.2	15.0441	33.0640	36.3256

Fuente: Elaboración propia

### 5.3.1. Resultados controlador proporcional

El cuadro 5.1 expone los diferentes porcentajes de error con los cuales es posible establecer un rango de variabilidad de los parámetros buscando obtener el mejor resultado. En la (*Figura 5.8*) se visualiza el resultado no deseado generado por el valor de las constantes de K1 y K2 igual a 2, los cuales presentan un margen de error teórico de 5,3063 disminuyendo el error presentado por el modelo matemático, pero frente a la gráfica presenta un comportamiento no deseado gracias al cambio constante de punto por lo que el error es mayor.

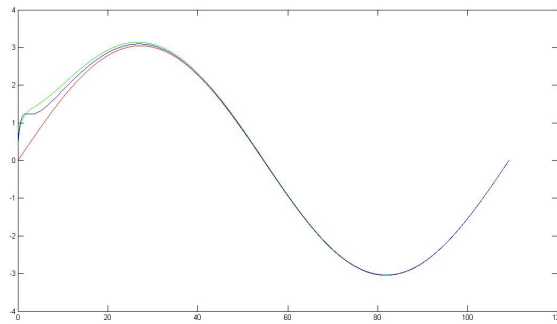
Figura 5.8: Resultados no deseados del controlador proporcional



Fuente: Elaboración propia

En la (Figura 5.9) muestra tres trayectorias donde se observa de color rojo la trayectoria deseada, de color verde la trayectoria presentada por la respuesta natural del sistema y de color azul la trayectoria utilizando el controlador proporcional con valores de ganancia de  $K_1=0.8$  y  $K_2=2$ , reflejando la disminución del error en la trayectoria en comparación de forma gráfica, por lo que es propuesto como el resultado óptimo encontrado en el control proporcional.

Figura 5.9: Resultados deseados del controlador proporcional



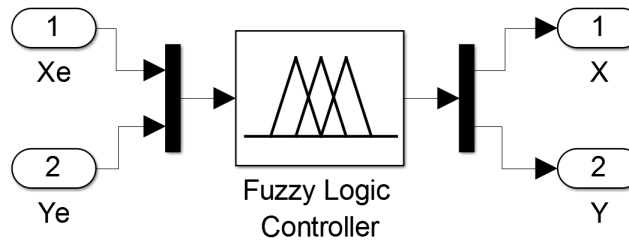
Fuente: Elaboración propia

## 5.4. Diseño de controlador difuso

El modelo cinemático observado anteriormente expone un sistema no lineal conformado por ecuaciones, ocasionando que el controlador adecuado para ser implementado sea el controlador difuso al estar fundamentado en la lógica difusa. Al aplicar la lógica difusa es necesario considerar la respuesta deseada con base a la entrada conocida en el sistema; utilizando los antecedentes vistos en los resultados en lazo abierto es planteado el controlador visto en la (Figura 5.10) por diagrama de bloques, donde se aplica

una ganancia en la entrada del bloque del controlador difuso el cual amplifica el error, causando una mejor respuesta al obtener un aumento de velocidad y una disminución del error.

Figura 5.10: Controlador difuso



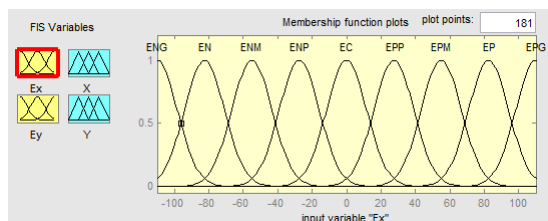
Fuente: Elaboración propia

El error es usado como fundamento en el diseño del controlador al ser la referencia para el comportamiento deseado, debido a eso fue necesario emplear dos entradas de error que representan la desviación de la trayectoria resultante en eje X y en el eje Y gracias a que el sistema utiliza las dos entradas para la respuesta del sistema en el plano XY ocasionado que el error dependa de la interacción entre los dos ejes.

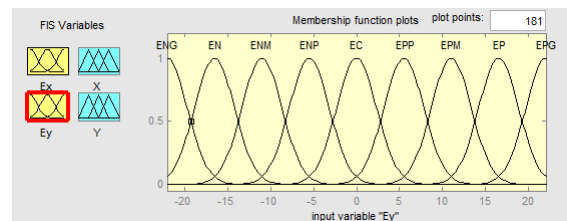
#### 5.4.1. Variable de entrada y de salida

Para establecer los parámetros de diseño de la señal de entrada de la trayectoria en el controlador se tiene en cuenta el comportamiento del sistema en lazo abierto donde se refleja la magnitud de la medida posible para el eje X y para el eje Y, encontrando así los rangos de entrada comprendidos entre  $-110$  y  $110$  respecto al eje X y  $-22$  hasta  $22$  correspondiente al eje Y. También, se aplican nueve (9) funciones de pertenencia proporcionando una gran exactitud en el seguimiento de la trayectoria y en el suavizado, al ser las funciones de pertenencia del tipo Gaussiano como se muestra en la (Figura 5.11) donde son expuestas las variables de entrada Xe y Ye correspondientes a las entradas del error en X y en Y.

Figura 5.11: Variables de entrada



(a) Error en coordenada X

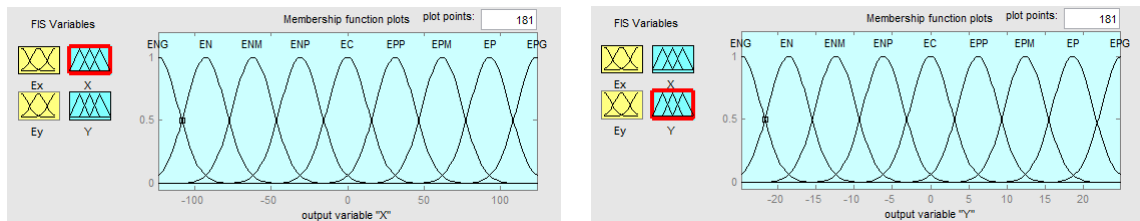


(b) Error en coordenada Y

Fuente: Elaboración Propia

Los parámetros de la señal de salida del controlador hacia el sistema implementan funciones de pertenencia similares a la de entrada, debido a la función de corrección del error deseado para el sistema; permitiendo establecer reglas de forma simple y eficiente donde se obtienen las salidas vistas en la (Figura 5.12) donde el rango de la coordenada de la salida X es comprendido entre  $-124$  y  $124$  y el de la salida Y se encuentra entre  $-24,8$  y  $24,8$ .

Figura 5.12: Variables de salida



(a) Salida del controlador coordenada X

(b) Salida del controlador coordenada Y

Fuente: Elaboración Propia

Por cuestiones de espacio fueron usadas abreviaciones para nombrar las diferentes funciones de pertenencia, donde: ENG corresponde al «Error negativo grande», EN es el «Error negativo», ENM corresponde al «Error negativo medio», ENP es el «Error negativo pequeño», EC correspondiente al «Error cero», EPP se llama al «Error positivo pequeño», EPM corresponde al «Error positivo medio», EP perteneciente al «Error positivo» y por último, EPG correspondiente al «Error positivo grande».

### 5.4.2. Reglas difusas

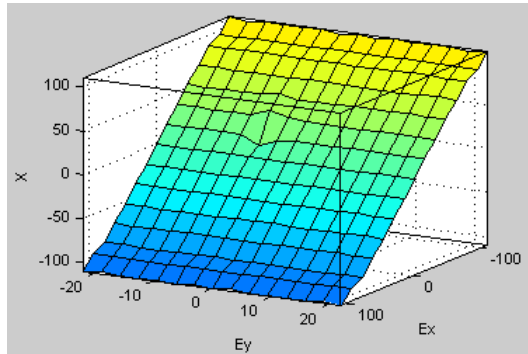
Las reglas difusas utilizadas para el control del sistema implementado son fundamentadas en el concepto de corrección del error ocasionando que la salida de las variables utilizadas sean opuestas a las variables de entrada, gracias a que el controlador se basa en el error de posición en las coordenadas  $(X, Y)$  permitiendo corregir la posición que se da siguiendo la trayectoria, de esta manera se obtiene el cuadro 5.2.

Cuadro 5.2: Reglas difusas implementadas

Función de pertenencia	ENG en Xe	EN en Xe	ENM en Xe	ENP en Xe	EC en Xe	EPP en Xe	EPM en Xe	EP en Xe	EPG en Xe
ENG en Ye	X=EPG Y=EPG	X=EP Y=EPG	X=EPM Y=EPG	X=EPP Y=EPG	X=EC Y=EPG	X=ENP Y=EPG	X=ENM Y=EPG	X=ENG Y=EPG	X=ENG Y=EPG
EN en Ye	X=EPG Y=EP	X=EP Y=EP	X=EPM Y=EP	X=EPP Y=EP	X=EC Y=EP	X=ENP Y=EP	X=ENM Y=EP	X=EN Y=EP	X=ENG Y=EP
ENM en Ye	X=EPG Y=EPM	X=EP Y=EPM	X=EPM Y=EPM	X=EPP Y=EPM	X=EC Y=EPM	X=ENP Y=EPM	X=ENM Y=EPM	X=EN Y=EPM	X=ENG Y=EPM
ENP en Ye	X=EPG Y=EPP	X=EP Y=EPP	X=EPM Y=EPP	X=EPP Y=EPP	X=EC Y=EPP	X=ENP Y=EPP	X=ENM Y=EPP	X=EN Y=EPP	X=ENG Y=EPP
EC en Ye	X=EPG Y=EC	X=EP Y=EC	X=EPM Y=EC	X=EPP Y=EC	X=EC Y=EC	X=ENP Y=EC	X=ENM Y=EC	X=EN Y=EC	X=ENG Y=EC
EPP en Ye	X=EPG Y=ENP	X=EP Y=ENP	X=EPM Y=ENP	X=EPP Y=ENP	X=EC Y=ENP	X=ENP Y=ENP	X=EPM Y=ENP	X=EN Y=ENP	X=ENG Y=ENP
EPM en Ye	X=EPG Y=ENM	X=EP Y=ENM	X=EPM Y=ENM	X=EPP Y=ENM	X=EC Y=ENM	X=ENP Y=ENM	X=ENM Y=ENM	X=EN Y=ENM	X=ENG Y=ENM
EP en Ye	X=EPG Y=EN	X=EP Y=EN	X=EPM Y=EN	X=EPP Y=EN	X=EC Y=EN	X=ENP Y=EN	X=ENM Y=EN	X=EN Y=EN	X=ENG Y=EN
EPG en Ye	X=EPG Y=ENG	X=EP Y=ENG	X=EPM Y=ENG	X=EPP Y=ENG	X=EC Y=ENG	X=ENP Y=ENG	X=ENM Y=ENG	X=EN Y=ENG	X=ENG Y=ENG

Fuente: Elaboración Propia

Entorno a las reglas difusas se obtiene la (*Figura 5.13*), donde se refleja la respuesta de las funciones de pertenencia resultantes de la aplicación de las reglas difusas observadas anteriormente exponiendo la magnitud de los valores obtenidos, donde el color azul oscuro corresponde al valor más bajo generado como respuesta de los errores más altos entre las entradas de error en X y en Y, junto con el color amarillo oscuro correspondiente a la respuesta más alta causado por el error mas bajo.

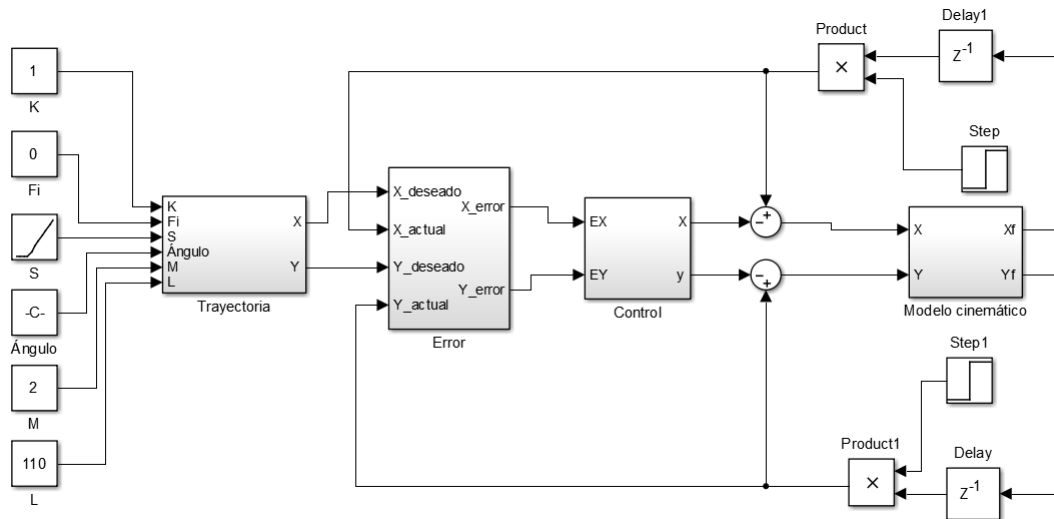
Figura 5.13: *Surface* del controlador difuso

Fuente: Elaboración propia

### 5.4.3. Resultados de control difuso

Para utilizar el controlador difuso se implementa el diagrama de bloques expuesto en la (Figura 5.14) donde se expone el retardo en la retro-alimentación de posición para hallar el error en las coordenadas entorno a la posición actual, la posición deseada proporcionada por la trayectoria, la inicialización obtenida del producto entre el pulso «Step» y la posición donde inicia en cero usando como referencia el tiempo de retardo. También se observa el posicionamiento actual retro-alimentando la entrada del sistema siendo sumado con el controlador proporcionando el cambio de posición entorno al error.

Figura 5.14: Diagrama del controlador difuso



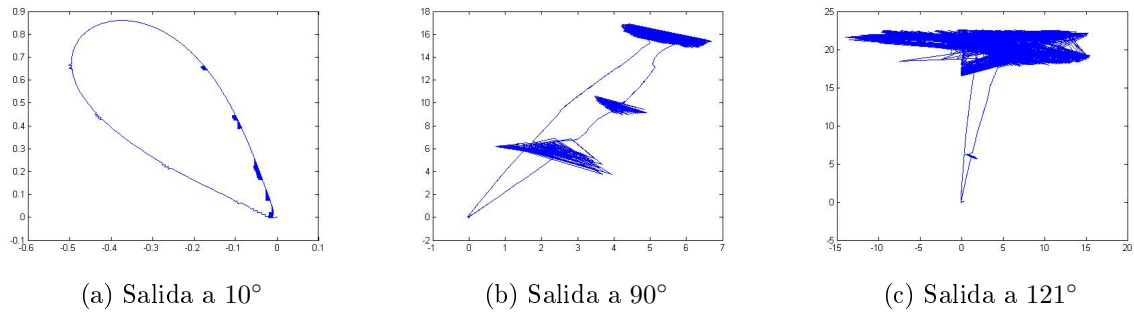
Fuente: Elaboración propia



### 5.4.3.1. Error resultante

Usando el sistema visto anteriormente se obtiene la (*Figura 5.15*) donde se muestra la salida del sistema con base a los ángulos de serpenteo de  $10^\circ$ ,  $90^\circ$  y  $121^\circ$ . La forma del error entorno al error en X y el error en Y, adquiere una forma ondulada que expone un rango de error para los  $10^\circ$  entre  $-0,5$  y  $0,5$  en X y de  $-0,1$  a  $0,9$  en Y. Para el ángulo de serpenteo de  $90^\circ$  el rango aumenta debido a que se encuentra entre los valores correspondientes para X entre  $-0,03$  y  $6,7$  junto con los valores del rango de Y entendidos entre  $-0,02$  y  $16,9$  demostrando un aumento en el error a corregir. El ángulo que presentó mayor repuesta fue la respuesta obtenida por el ángulo de serpenteo de  $121^\circ$  debido a que se encuentra entre el rango en X de  $-14$  y  $15,5$  junto con el rango en Y de  $-0,02$  y  $22,5$ .

Figura 5.15: Salida controlador difuso



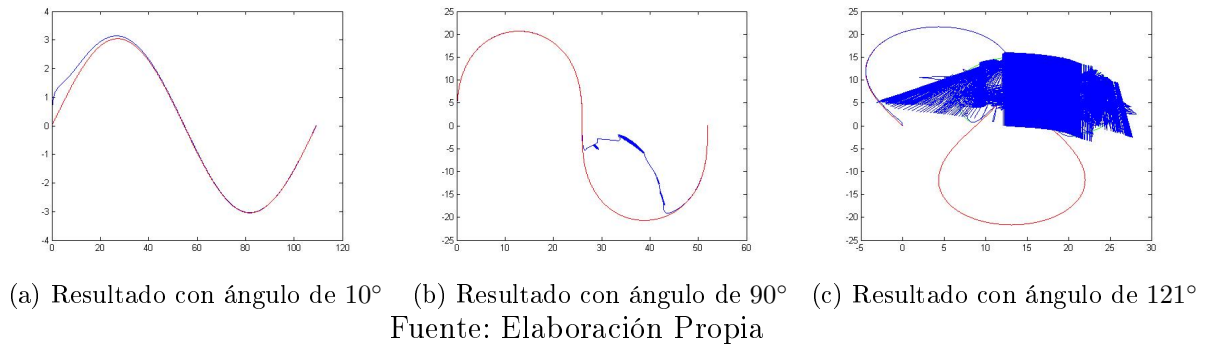
Fuente: Elaboración propia

La (*Figura 5.15*) refleja las limitaciones en el ángulo de serpenteo presentados gracias al número de módulos estudiados, por lo cual se encuentra mayor margen de error a medida que aumenta el ángulo, el cual excede las limitaciones físicas y teóricas planteadas para un módulo, por lo que para mayor ángulo de serpenteo es necesario usar mayor número de módulos.

### 5.4.3.2. Trayectoria resultante

Utilizando el diagrama de bloques del control difuso (visto en la *Figura 5.14*), se encuentra la (*Figura 5.16*) donde se muestra de color azul la salida del sistema utilizando el controlador y de color rojo la trayectoria de entrada al sistema donde se refleja una corrección mínima del error de la trayectoria, debido a esto es necesario complementar el controlador para mejorar la respuesta. Además, se presenta un descontrol frente a ángulos mayores de  $10^\circ$  como es el ángulo de  $90^\circ$  y  $121^\circ$  que presentan mayor error y superan el ángulo de serpenteo mayor al recomendado de  $30^\circ$ .

Figura 5.16: Resultado controlador difuso

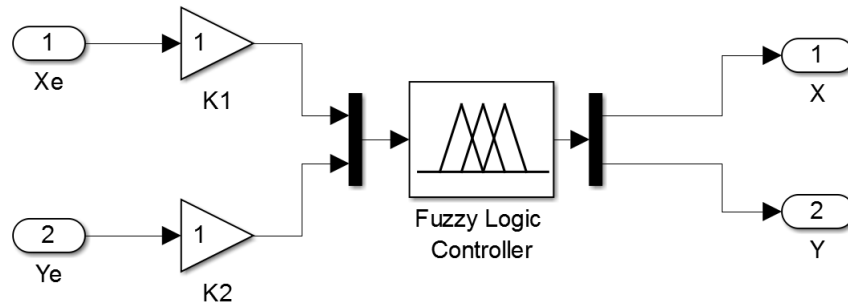


En la (Figura 5.16) es expuesto el aumento del margen de error frente al aumento del ángulo de serpenteo ocasionado por las limitaciones presentadas por los módulos. El descontrol presentado por el sistema en el transcurso de la trayectoria es atribuido no sólo a las limitaciones presentadas por el módulo como se mencionó anteriormente, debido a la necesidad presentada de controlar cada módulo frente al desplazamiento por lo que al aumentar el número de módulos deben aumentar el número de controladores.

## 5.5. Diseño de controlador proporcional-difuso

Para realizar el diseño del controlador proporcional-difuso se realiza la estructura expuesta en la (Figura 5.17) donde las entradas del controlador son amplificadas por dos variables ( $K_1$  y  $K_2$ ), las cuales amplifican o disminuyen el error obteniendo una respuesta más rápida o lenta de acuerdo a la necesidad presentada.

Figura 5.17: Controlador proporcional-difuso



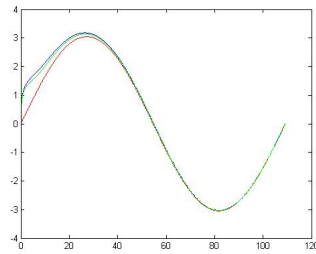
Fuente:Elaboración propia

Evaluando el comportamiento de las variables frente a la respuesta del controlador proporcional-difuso se utilizaron varios valores para  $K_1$  y  $K_2$  variando el ángulo de serpenteo, obteniendo las siguientes gráficas donde la trayectoria corresponde al color

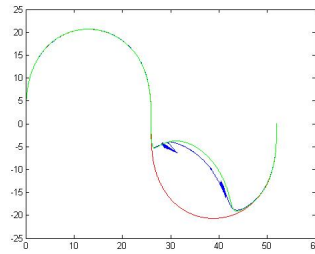
rojo, la respuesta del sistema de color verde y el resultado del sistema con el controlador de color azul.

- Respuestas con  $K_1=1.5$ ,  $K_2=0.8$  y variación de ángulo de serpenteo

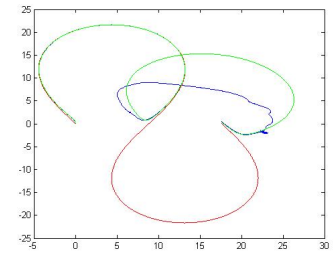
Figura 5.18: Respuesta difusa con  $K_1=1.5$  y  $K_2=0.8$



(a) Respuesta a  $10^\circ$



(b) Respuesta a  $90^\circ$

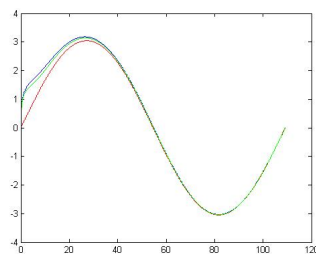


(c) Respuesta a  $121^\circ$

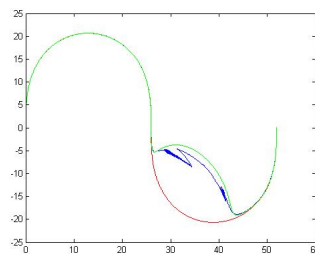
Fuente: Elaboración Propia

- Respuesta con  $K_1=1.1$ ,  $K_2=0.8$  y variación de ángulo de serpenteo

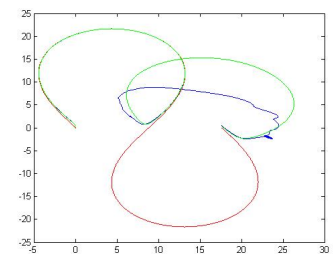
Figura 5.19: Respuesta difusa  $K_1=1.1$  y  $K_2=0.8$



(a) Respuesta a  $10^\circ$



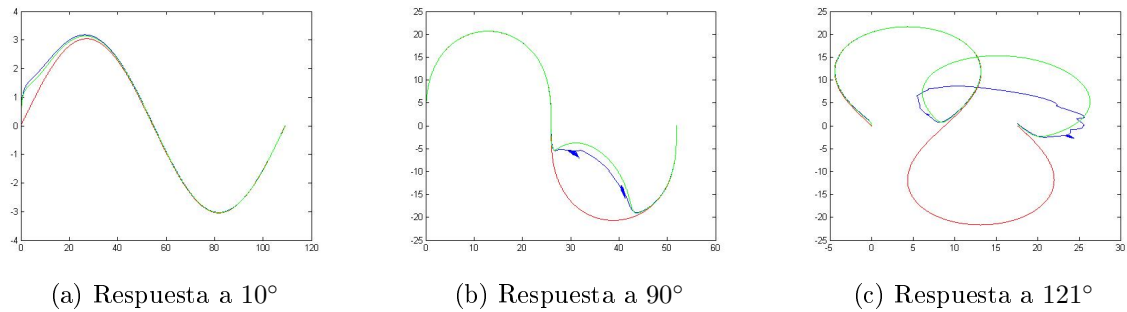
(b) Respuesta a  $90^\circ$



(c) Respuesta a  $121^\circ$

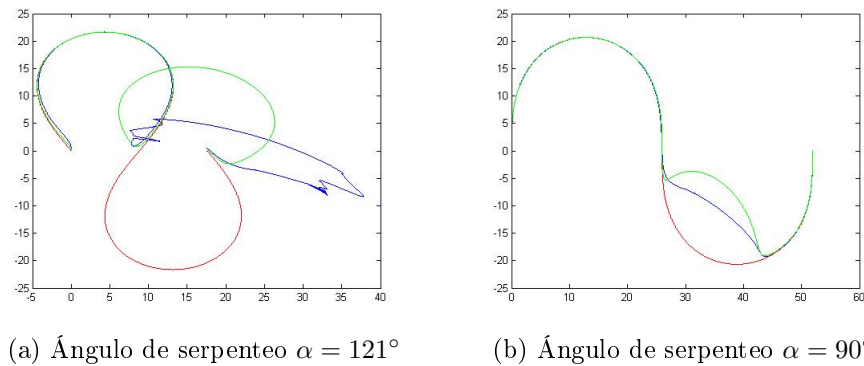
Fuente: Elaboración Propia

- Respuesta con  $K_1=0.8$ ,  $K_2=0.8$  y variación de ángulo de serpenteo

Figura 5.20: Respuesta difusa  $K1=0.8$  y  $K2=0.8$ 

Fuente: Elaboración Propia

Las gráficas anteriores muestran: la trayectoria de color rojo, la respuesta del sistema en lazo abierto de color verde y la respuesta del sistema con el controlador difuso de color azul mostrando el mejoramiento en la respuesta entorno al cambio de los parámetros de  $K1$  y  $K2$ . En las variaciones expuestas se encuentra una mejor reacción con los valores de  $K1=0.8$  y  $K2=0.8$  frente a los valores cercanos al ángulo de serpenteo mínimo y máximo permitidos por las cualidades del sistema, la variación de las variables  $K1$  y  $K2$  permiten deducir que a menor valor hay una mejor respuesta, cosa que no es del todo cierta al fijarse en la (Figura 5.21) donde se expone una respuesta no deseada del sistema cuando el valor de  $K1=0.2$  y  $K2=0.8$  con ángulo de serpenteo de  $121^\circ$  y una respuesta deseada con los valores de  $K1=0.2$  y  $K2=0.8$  con ángulo de serpenteo de  $90^\circ$  ocasionando que los valores no sean adecuados para la aplicación debido a la necesidad de cambio constante del ángulo de serpenteo en la aplicación y la falta de constancia en la corrección del error, por lo que es necesario verificar las cualidades físicas del robot.

Figura 5.21: Respuesta difusa  $K1=0.2$ ,  $K2=0.8$ 

Fuente: Elaboración Propia

Las ganancias aplicadas demuestran claramente un rango de acción, gracias a esto es necesario definir limitaciones de diseño el cual por efecto de los motores comprende

un rango para el ángulo de serpenteo máximo entre  $0^\circ$  y  $90^\circ$  junto con un rango en el ángulo de serpenteo de  $5^\circ$  hasta  $30^\circ$  correspondiente a las limitaciones de diseño mecánico para poder generar un desplazamiento adecuado. Evaluando las limitaciones se seleccionó el ángulo de serpenteo de  $10^\circ$  para evaluar los valores de las constantes del controlador, obteniendo el cuadro 5.3 donde es evaluado el error promedio en el eje X denotado  $E_x$ , el error promedio en Y denotado  $E_y$  y el error total denotado E, siendo obtenido por medio de la ecuación 5.4.

Cuadro 5.3: Variación de constantes del controlador proporcional- difuso

K1	K2	$E_x$ (%)	$E_y$ (%)	E (%)
2	2	4.679	5.8088	7.4589
2	1.5	2.2918	6.7859	7.1624
2	1.1	34.8878	151.4378	155.4045
2	0.8	168.8941	207.7802	267.7645
2	0.2	1.1256	15.151	15.1927
1.5	2	0.8381	6.2057	6.2620
1.5	1.5	1.5036	2.8558	3.2274
1.5	1.1	1.5117	3.8104	4.0993
1.5	0.8	1.4621	4.9858	5.1957
1.5	0.2	1.4274	17.9622	18.0188
1.1	2	0.6750	8.842	8.8677
1.1	1.5	1.976	3.1734	3.7383
1.1	1.1	2.013	4.3607	4.8029
1.1	0.8	1.9613	5.7489	6.0742
1.1	0.2	1.91	20.7673	20.8549
0.8	2	1.3573	5.9846	6.1365
0.8	1.5	2.6249	3.0639	4.0345
0.8	1.1	2.8362	4.411	5.2441
0.8	0.8	2.7457	5.3915	6.0503
0.8	0.2	2.6701	19.6124	19.7933
0.2	2	10.5289	2.0387	10.7244
0.2	1.5	10.954	2.8819	11.3267
0.2	1.1	10.941	3.9562	11.6343
0.2	0.8	11.7051	5.8388	13.0805
0.2	0.2	13.0816	26.4647	29.5213

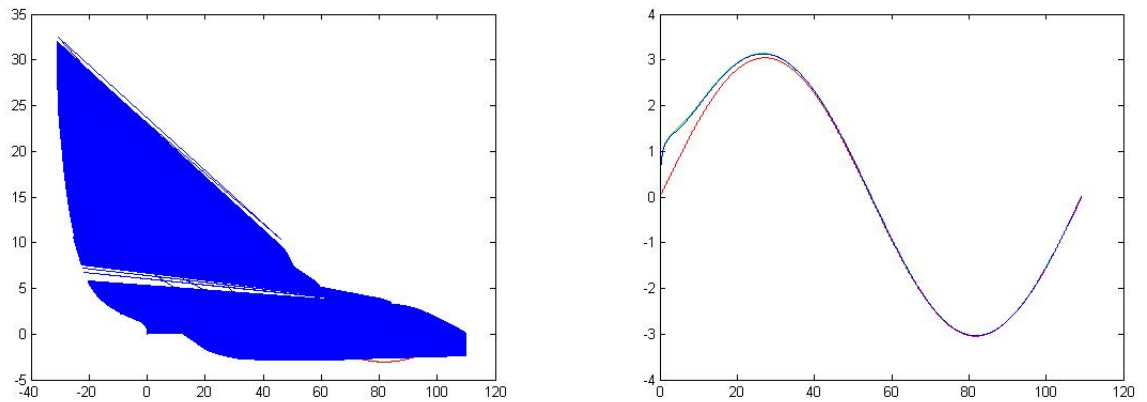
Fuente: Elaboración propia

### 5.5.1. Resultados del controlador proporcional-difuso

En el cuadro 5.3 se visualizan los resultados del error entorno a la variación de las ganancias K1 y K2, utilizando los resultados hallados es encontrada la (*Figura 5.22*) donde se muestra el resultado que presentó mayor error, junto con el resultado que no

mostró cambio alguno y también con el presentado por la respuesta natural del sistema. Aunque el comportamiento del controlador cambia entorno a la variación y combinación de las ganancias, observando los resultados es posible encontrar las ganancias de  $K_1$  comprendidas entre 1.1 hasta 2 y de  $K_2$  entre 1.1 hasta 0.8, presentan un margen de error no deseado al ser similar o mayor al generado por la respuesta natural de sistema.

Figura 5.22: Resultados no deseados del controlador proporcional-difuso

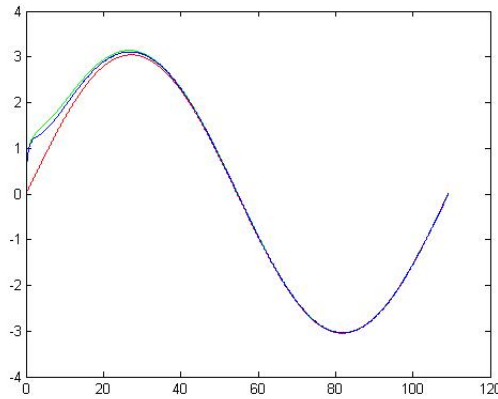
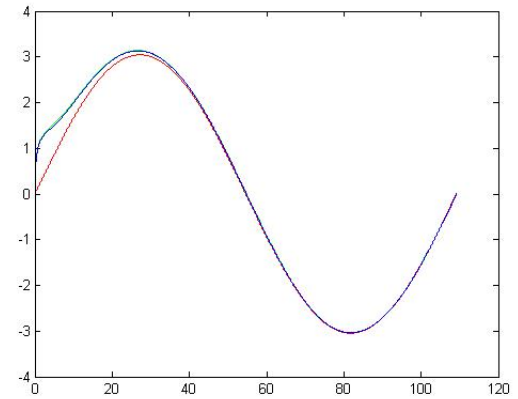


(a) Respuesta de error máximo ( $K_1=2$  y  $K_2=0.8$ ) (b) Respuesta similar a la respuesta similar del sistema ( $K_1=1.1$  y  $K_2=1.1$ )

Fuente: Elaboración propia

Así como se encontraron resultados no deseados, también fueron encontrados resultados deseados al disminuir el error del sistema frente a la trayectoria deseada como es expuesto en la (Figura 5.23), mostrando los valores de las ganancias que no generan cambios en el sistema y los valores que generaron una mejor respuesta disminuyendo el error. Observando los resultados presentados es posible proponer los valores de las ganancias de  $K_1$  y  $K_2$  entre 1.5 y 1.1.

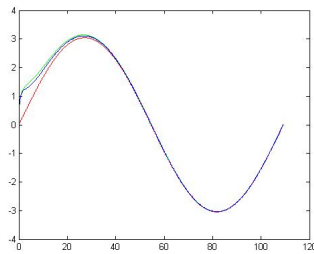
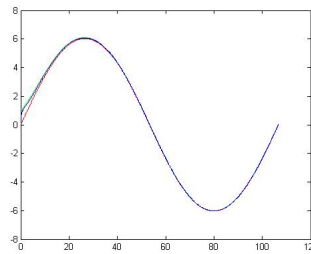
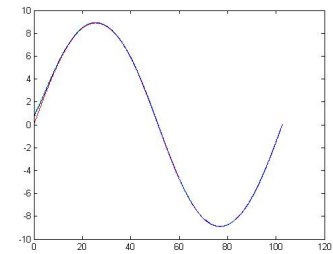
Figura 5.23: Resultados deseados del controlador proporcional-difuso

(a) Mejor respuesta ( $K_1=1.5$  y  $K_2=1.5$ )(b) Respuesta similar a la respuesta similar del sistema ( $K_1=1.1$  y  $K_2=1.1$ )

Fuente: Elaboración propia

Al usar los valores obtenidos que generan menor error según el cuadro 5.3, son implementadas las ganancias de  $K_1=1.5$  y de  $K_2=1.5$  para los ángulos de serpenteo de  $10^\circ$ ,  $20^\circ$  y  $30^\circ$  obteniendo la (Figura 5.24) visualizando los resultados donde se usa el color rojo para la señal de trayectoria, el color verde para la respuesta natural del sistema y de color azul la respuesta obtenida con el controlador.

Figura 5.24: Mejor respuesta variando el ángulo de serpenteo

(a) Ángulo de serpenteo de  $10^\circ$ (b) Ángulo de serpenteo  $20^\circ$ (c) Ángulo de serpenteo  $30^\circ$ 

Fuente: Elaboración propia

Entorno a la (Figura 5.24) es realizado el cuadro donde el error promedio en X es denotado como  $E_x$ , el error promedio en Y es denotado como  $E_y$  y el error promedio del sistema es denotado como  $E$ . Analizando los resultados obtenidos se puede ver la disminución del error generado por el sistema, encontrando un menor margen de error proporcional al cambio del ángulo, por lo que mejora la respuesta a medida que se realiza este cambio. Además, varía el comportamiento en el transcurso de la trayectoria,

generando un error como se presenta en los ángulos de  $90^\circ$  y de  $121^\circ$  vistos en la (Figura 5.4) correspondiente a la respuesta del modelo cinemático.

Cuadro 5.4: Variación del error mínimo en torno al ángulo

Ángulo de serpenteo	Ex (%)	Ey (%)	E (%)
$10^\circ$	1.5036	2.8558	3.2274
$20^\circ$	1.6178	2.1614	2.6998
$30^\circ$	1.3598	1.3755	1.9341

Fuente: Elaboración propia

Observando los resultados obtenidos de forma empírica se demuestra la posibilidad de mejorar el sistema usando métodos de optimización más avanzados. Los métodos de optimización mejorarían el sistema al plantear de forma precisa los valores de ganancia que permiten disminuir el error de forma deseada acorde a la necesidad planteada.

## 5.6. Comparación de resultados

Observando los resultados encontrados de los diferentes controladores estudiados anteriormente se realiza el cuadro 5.5, donde son expuestos los errores en el eje X y en el eje Y junto con su error promedio resultante. Además, se resalta la disminución del error frente al cambio de controlador, por lo que el controlador proporcional-difuso presenta una disminución óptima del margen de error.

Cuadro 5.5: Comparación de controladores

Controlador	Error en X (%)	Error en Y (%)	Error Total (%)
Controlador proporcional	4.0140	6.7073	7.8166
Controlador difuso	2.5622	6.2257	6.7323
Controlador proporcional-difuso	1.5036	2.8558	3.2274

Fuente: Elaboración propia

Los controladores estudiados presentan diferentes comportamientos con base a los cambios de parámetros presentados como es la variación del ángulo de serpenteo. Por lo tanto, se hizo el cuadro 5.6 que presenta ventajas y desventajas de los controladores según el cambio de parámetros.



Cuadro 5.6: Ventajas y desventajas de los controladores diseñados

Controlador	Ventajas	Desventajas
Controlador proporcional	Presenta mayor velocidad de procesamiento y disminución adecuada del error de forma gráfica.	Presenta un aumento en el margen de error según la variación del ángulo de serpenteo.
Controlador difuso	Presenta un comportamiento adecuado y constante en ángulos de serpenteo adecuados para el sistema	Tiende a descontrolarse frente a ángulos de serpenteo inadecuados para el número de módulos estudiados.
Controlador proporcional-difuso	El comportamiento obtenido refleja un comportamiento adecuado tanto en el ángulo de serpenteo recomendado para su utilización, como en las variaciones abruptas que podría sufrir.	A pesar que presenta mejor comportamiento a comparación de los dos controladores aún así tiende a descontrolarse frente a ángulos de serpenteo inadecuados pero con menor margen de error que el presentado.

Fuente: Elaboración propia

## Capítulo 6

# CONCLUSIONES Y TRABAJOS FUTUROS

### 6.1. Conclusiones

En este trabajo son presentados los principales parámetros de diseño para realizar un controlador diseñado entorno al modelo matemático de un robot ápedo basándose en la estructura normalmente usada para el estudio de locomoción de este tipo de robot, exponiendo los factores más importantes comúnmente encontrados en los robots ápedos como son: el ángulo de serpenteo, número de módulos, longitud del módulo, entre otros. Por lo que se puede decir que es contestada de forma adecuada la pregunta problema «¿Cómo se podría diseñar un sistema de control adecuado para los robots ápedos considerando los parámetros que se deben tener en cuenta para su correcto funcionamiento?». ».

Una de las partes importantes del modelo matemático implementado en este trabajo es el enfocado en el movimiento sinusoidal característico de los robots del tipo ápedo, en el cual es la basado el movimiento de serpenteo que describe de forma precisa el movimiento del robot. Este movimiento de serpenteo describe de forma adecuada la trayectoria de robot, que comprende un espacio de trabajo desde 0 hasta la longitud total del robot denotada anteriormente como  $l$ . Contemplando múltiples factores que interfieren en la trayectoria del robot en el espacio continuo, describiendo el movimiento realizado desde el punto de inicio encontrado en las coordenadas  $(0, 0)$  hasta las coordenadas finales. Buscando hallar el posicionamiento de las articulaciones formando la trayectoria, es necesario realizar el análisis en el espacio discreto debido a que se aproxima al posicionamiento de los módulos del robot ápedo.

Realizando énfasis en el modelo matemático entorno al posicionamiento y movimiento del robot se realiza el modelo cinemático comprendido por la cinemática directa fundamentada en la matriz de «*Denavit-Hartenberg*», que ayuda a encontrar el posicionamiento con base a la longitud y los ángulos formados por el robot obteniendo las

coordenadas de posicionamiento demostrando una mayor facilidad en la implementación al considerar las cualidades físicas del robot. También el modelo cinemático contempla la cinemática inversa que presenta mayor dificultad al tener que entender las cualidades físicas del robot junto con el espacio de trabajo del robot y movimiento realizado.

El modelo matemático no contempla el análisis dinámico del sistema debido a que la complejidad presentada requeriría un trabajo de grado enfocado en ese análisis dinámico entorno a un diseño preseleccionado y por lo tanto requiere una mayor cantidad de tiempo para su realización, por lo que sólo se analizó un módulo y la capacidad de soportar un impacto con el peso completo del robot, debido a que es necesario que cada módulo resista el peso completo del robot. Además, se realizó el respectivo estudio de material comúnmente usado, donde se destacan el aluminio y el ABS, al presentar deformación mínima y uniforme en la estructura frente a los demás materiales.

Al aplicar el modelo cinemático que contempla la cinemática inversa se encuentra la necesidad de aplicar la trayectoria obtenida del movimiento de serpenteo derivada del movimiento sinusoidal característico del robot. Al utilizar la trayectoria se asegura el posicionamiento y desplazamiento adecuado del robot, debido a que contempla las cualidades y limitaciones físicas presentadas.

El modelo cinemático implementado presenta un comportamiento no lineal generando la necesidad de implementar un controlador difuso adecuado para sistemas no lineales. Al usar un controlador difuso fue requerido utilizar múltiples funciones de pertenencia causando el aumento en la exactitud de la respuesta del sistema permitiendo un mejor control y análisis del sistema. Pero el controlador presentó una corrección de error mínimo teniendo la necesidad de complementar el controlador difuso.

El controlador proporcional-difuso expone una relación entre la variación de las ganancias implementadas junto al cambio del margen de error al aumentar la velocidad de respuesta del controlador. La variación de las constantes del controlador presentan un aumento o disminución del error, ocasionado que la selección inadecuada de las constantes generen un comportamiento inadecuado del sistema, como también una selección adecuada de ganancias logra presentar una mejora en la respuesta del sistema mostrando un mejor comportamiento frente al error. Para encontrar los mejores valores de ganancia para minimizar el error de forma óptima es necesario aplicar métodos avanzados de optimización.

## 6.2. Trabajos futuros

Como fue mencionado anteriormente el modelo dinámico del sistema del robot ápodio presenta cierto margen de dificultad con el cual podría ser propuesto un trabajo de grado enfatizando en el desarrollo del modelo dinámico, considerando los diferentes factores

presentados como son: la locomoción, el material y la estructura, con el fin de realizar un diseño funcional para la futura fabricación e implementación del controlador planteado.

Planteando el diseño entorno a un modelo dinámico como es argumentado en el ANEXO D se podría plantear un nuevo modelo cinemático con el número total de módulos mayores a 3, con el fin que el modelo cinemático sea cercano a la linealidad y por ende sea posible diseñar un controlador PID, el cual podría complementar el sistema de control realizando un controlador PID-fuzzy.

# Bibliografía

- [1] Jessica Hazdey Barreto Paz and Lizet Tatiana Medina Ramírez. Simulador de trayectorias 2d para un robot tipo serpiente. 2010.
- [2] Ferdinand Pierre Beer, E Russell Johnston, José Antonio Gómez-Jurado García, and Juan de la Rubia Pacheco. Mecánica vectorial para ingenieros. Technical report, McGraw-Hill, 1990.
- [3] STEEVE ERASMO TOLEDO CHOJOLÁN. Diseño de controladores pid en tiempo discreto, y análisis de respuesta utilizando herramientas computacionales. Master's thesis, Universidad de San Carlos de Guatemala, 2007.
- [4] Jeraldinne Molleda Miguel Osorio Junior Vivas Noel Heredia Claudia Montero, Duglenys Guanipa. resumen control cascada.
- [5] Copyright. Towerpro sg-5010 servo. [www.servodatabase.com/servo/towerpro/sg-5010](http://www.servodatabase.com/servo/towerpro/sg-5010), 2016. Accessed: 2016-05-11.
- [6] Ro Crespi, André Badertscher, André Guignard, and Auke Jan Ijspeert. Swimming and crawling with an amphibious snake robot, 2005.
- [7] Howie Choset David Rollinson. Virtual chassis for snake robots.
- [8] Richard C Dorf, Robert H Bishop, Sebastián Dormido Canto, Raquel Dormido Canto, and Sebastián Dormido. *Sistemas de control moderno*. Pearson Prentice Hall, 2005.
- [9] Isin Erenoglu, Ibrahim Eksin, Engin Yesil, and Mujde Guzelkaya. An intelligent hybrid fuzzy pid controller. In *European Conference on Modelling and Simulation*, pages 62–67, 2006.
- [10] Juan González Gómez. Principales líneas de investigación en robots tipo ápodos. 2002.
- [11] Juan González Gómez. Robótica modular y locomoción: Aplicación a robots ápodos. *Universidad Autónoma de Madrid, Thesis*, 2008.

- [12] Juan González, Ivan González, and Eduardo Boemo. Alternativas hardware para la locomoción de un robot ápodo. *III Jornadas sobre Computación Reconfigurable y Aplicaciones, Escuela Politécnica Superior, Universidad Autónoma de Madrid*, 2003.
- [13] Juan González-Gómez, Ivan González, Francisco J Gómez-Arribas, and Eduardo Boemo. Evaluación de un algoritmo de locomoción de robots Ápodos en diferentes procesadores embebidos en fpga. *V Jornadas de Computación Reconfigurable y Aplicaciones. JCRA*, pages 1–8, 2005.
- [14] René Jiménez Madrigal. Diseño de un controlador lógico difuso, aplicado al control de posición de un servomotor de cd, usando un algoritmo genético. 2007.
- [15] Hadi Kalani, Alireza Akbarzadeh, and Hossein Bahrami. Application of statistical techniques in modeling and optimization of a snake robot. *Robotica*, 31(4):623–641, 07 2013. Copyright - Copyright © Cambridge University Press 2012; Última actualización - 2013-05-24.
- [16] Soon Ju Kang Laxmisha Rai. Multi-thread based synchronization of locomotion control in snake robots.
- [17] Kevin Lipkin, Isaac Brown, Aaron Peck, Howie Choset, Justine Rembisz, Philip Gianfortoni, and Allison Naaktgeboren. Differentiable and piecewise differentiable gaits for snake robots. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1864–1869. IEEE, 2007.
- [18] Samuel Muñoz González et al. Análisis de la cinemática inversa y minimización de errores mediante algoritmos de optimización para un robot hiper-redundante. 2013.
- [19] Katsuhiko Ogata. *Ingeniería de control moderna*. Pearson Educación, 2010.
- [20] Kevin M Passino, Stephen Yurkovich, and Michael Reinfrank. *Fuzzy control*, volume 42. Citeseer, 1998.
- [21] José Alberto Mendez Polanco, Erika Amaro Camargo, Angelica Munoz Melendez, Eduardo F Morales Manzanares, and Luis Enrique Erro No. Diseño y construcción de un robot apodo.
- [22] Auke Jan Ijspeert Ro Crespi. Online optimization of swimming and crawling in an amphibious snake robot, 2008.
- [23] Andrés San Millán Rodríguez. Diseño, construcción y control de una serpiente robótica. In *Creatividad, descubrimiento y futuro: I Congreso Nacional de Investigación en Grado INVESGRADO 2011, Albacete, Junio de 2011*, pages 1385–1399. Universidad de Castilla-La Mancha, 2011.

- [24] Andres San Millan Rodriguez. Diseño, construcción y control de una serpiente robótica. In *Creatividad, descubrimiento y futuro: I Congreso Nacional de Investigación en Grado INVESGRADO 2012, Albacete, 11 de mayo de 2012*, pages 1385–1399. Universidad de Castilla-La Mancha, 2012.
- [25] Yansong Shan and Yoram Koren. Design and motion planning of a mechanical snake. *Systems, Man and Cybernetics, IEEE Transactions on*, 23(4):1091–1100, 1993.
- [26] Stephen Tully, George Kantor, Marco A. Zenati, Howie Choset, Stephen Kantor, George Zenati, A. Marco, Stephen Tully, George Kantor, Marco A. Zenati, and Howie Choset. Highly articulated snake robot.
- [27] María Urdaneta and Roque Saltaren. Estudio del sistema de tracción de una estructura robótica para tuberías en sistemas de extracción petrolera con bes y bcp. *XXIX Jornadas CEA*, 2008.
- [28] Gholamreza Vossoughi, Hodjat Pendar, Zoya Heidari, and Saman Mohammadi. Assisted passive snake-like robots: conception and dynamic modeling using gibbs-appell method. *Robotica*, 26(3):267–276, 05 2008. Copyright - Cambridge University Press; Última actualización - 2010-06-08.
- [29] Cornell Wright, Aaron Johnson, Aaron Peck, Zachary Mccord, Allison Naaktgeboeren, Philip Gianfortoni, Manuel Gonzalez-rivero, Ross Hatton, and Howie Choset. Design of a modular snake robot.
- [30] Natalie Segura Velandia y Oswaldo Rivera Rinco. Control fuzzy para bi-rotor helicoptero.

# ANEXOS

## Anexo A

Second International Conference on Advanced Mechatronics, Design, and Manufacturing Technology - AMDM 2014

### DISEÑO Y CONSTRUCCIÓN DE UN ROBOT GUSANO TIPO EXPLORADOR

Designing and building of an explorer worm type bot

#### RESUMEN

En el presente artículo van a ver una propuesta de diseño de un robot explorador tipo ápodo con su respectivo análisis enfatizado en el desplazamiento y esfuerzos que surgen en éste. El movimiento zigzagante, característica de este tipo de robot, al proporcionar un movimiento rectilíneo que se puede adaptar fácilmente a los cambios de altura como de terreno, como los escombros y tuberías por lo que se usa para la búsqueda de minerales y poder encontrar el área navegable de la zona.

**PALABRAS CLAVE:** Ápodo, arduino nano, movimiento zigzagante, robot explorador, servomotor.

#### ABSTRACT

*In the present article you're going to see a proposal of design of an explorer robot "legless" type, with its respective review, emphasized in the effort and displacements which come out in this one. The rectilinear movement that can easily adapt to the height changes as well as terrain, like rubbles and pipes, for which is used to the reaching of minerals and to find out the navigable area of the zone*

**KEYWORDS:** Arduino nano, Explorer robot, legless, servomotor, Zigzag movement.

#### JONATHAN BERMUDEZ

Estudiante de Ingeniería Mecatrónica  
Universidad Piloto de Colombia  
Jbermudez42@upc.edu.co

#### TATIANA MARTINEZ

Estudiante de Ingeniería Mecatrónica  
Universidad Piloto de Colombia  
Cmartinez99@upc.edu.co

#### RUBEN HERNANDEZ

Ingeniero Electrónico, Ph.D.  
Estudiante de doctorado  
Profesor de ingeniería mecatrónica  
Universidad Piloto de Colombia  
Oscar-aviles@upc.edu.co

#### OSCAR FERNANDO AVILES

Ingeniero Electrónico, Ph.D.  
Profesor de ingeniería mecatrónica  
Universidad Piloto de Colombia  
Oscar-aviles@upc.edu.co

### 1. INTRODUCCIÓN

Los robot exploradores del tipo ápodo son una clase poco explorada a comparación de otras clases de robot exploradores como son el tipo oruga o el tipo zoomórfico, esta clase de robot surge de la necesidad de explorar lugares estrechos como túneles y grietas en las cuales las otras clases de robots exploradores no son tan eficaces, debido a que el tamaño de estos robots es muy grande y no puede ser utilizado para estos usos. Frente a esta problemática, los robots ápodos fueron una solución debido a que está compuesto por celdas unitarias con las que éste robot puede adaptarse a los distintos cambios que surgen en los terrenos a explorar.

Se buscó plantear un robot ápodo con el cual se pueda analizar el movimiento zigzagante o sinusoidal que caracteriza principalmente a esta clase de robot. Este movimiento se realiza en dos dimensiones con la que se distribuyen los puntos de apoyo en el tiempo formando una onda sinusoidal y generar un punto de control a través de un sensor de posición, el cual nos ayuda a que con la detección de algún tipo de objeto el robot se detenga para evitar algún tipo de choque. Además, emite una señal de aviso para saber cuándo el robot se detiene o cuando está explorando.

Los robots gusano o tipo ápodo se caracterizan principalmente por su facilidad a la hora de ejecutar su

movimiento, sin embargo, su desplazamiento proporciona algunas dificultades debido a restricciones propias del modelo o diseño cinemático (material) y además el esfuerzo de cada uno de los motores a la hora de la locomoción del robot. Para evitar esto se decidió modificar la forma de la celda unitaria, la cual permite mejorar el punto de apoyo aumentando la fricción entre el terreno y el robot proporcionando un desplazamiento adecuado, también se escogió aumentar el número de motores para posicionar un motor en cada punto de apoyo para distribuir mejor la fuerza de cada motor y eliminar la necesidad de motores con mayor fuerza, con estos ajustes mejorar el diseño y el análisis del movimiento del ápodo.

El diseño del robot gusano desarrollado se basó principalmente en una pieza unitaria similar a las encontradas en [3] y en [5], la cual permite mayor desplazamiento y locomoción gracias a los puntos de apoyo que presenta. Además, se incorporó una forma semi-triangular para aumentar la fricción y desplazamiento del robot. También, el robot gusano se diseñó de esta forma con el objetivo de cumplir distintas funciones como explorar grietas pequeñas o usarlo en la industria minera y de gases al igual que los robots presentados en [2], [3] y en [5].

Todo esto se realizó con el fin de poder diseñar y analizar esta clase de robot, el cual está enfocado a la exploración



y análisis de terrenos a través de sensores indicando al usuario en tiempo real el estado del terreno, para asegurar el recorrido o posición de un elemento con propiedades que el usuario requiera.

Este artículo después de la introducción está dividido de la siguiente manera: la sección 2 explica el diseño y las distintos factores que se analizaron para evaluar las cualidades de este, en la sección 3 explicamos las condiciones que se debieron tener en cuenta para realizar la programación y el objetivo de este sistema, luego encontramos la sección 4 se realizó el análisis de resultados para llegar así a la sección 5 que corresponde a las conclusiones que se pudieron obtener con el resultado y análisis realizados y finalizamos con la sección 6 de la bibliografía.

## 2. ESTRUCTURA MECÁNICA

El diseño del robot fue realizado para proporcionar un movimiento zigzagueante a través de los cambios de posición de un servomotor. Para poder ocasionar su desplazamiento se tuvo en cuenta la fricción en los puntos de apoyo principales del robot al momento de impulsarse.

### 2.1. Rodamiento

Un rodamiento es una pieza mecánica diseñada para disminuir la fricción, está compuesto por dos placas que encierran unos cuerpos rodantes como bolas o rodillos los cuales permiten un desplazamiento independiente entre las placas.

### 2.2. Cabeza

Esta pieza fue diseñada para mantenerse unida al suelo por lo que posee aristas rectas, posee dos agujeros a cada lado el cual se incrusta en un lado el tornillo y el otro un rodamiento, al ser una pieza unitaria este se requiere estos agujeros.

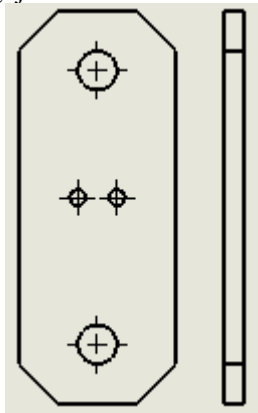


Imagen 2. Pieza de unión de la cabeza

### 2.3. Cuerpo

Esta pieza está basada en la misma de la cabeza pero con una modificación de inclinación de la arista para lograr una figura semi-triangular, con la que se proporciona mayor desplazamiento y flexión sobre la superficie.

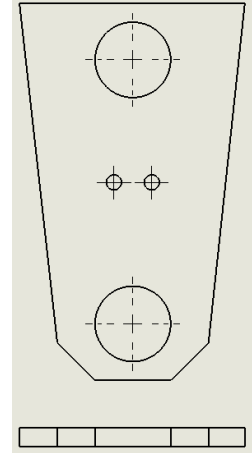


Imagen 3. Pieza de unitaria del cuerpo

### 2.4. Ensamble

Se realizó un ensamble de forma directa entre las piezas y los motores usando en ciertos puntos los rodamientos como punto de apoyo para esto, dando como resultado el ensamble mostrado en la imagen 4.

Este diseño físico del robot proporciona un movimiento zigzagueante a través de los cambios de posición de un servomotor, para poder ocasionar su desplazamiento se tuvo en cuenta la fricción en los puntos de apoyo principales del robot al momento de impulsarse.

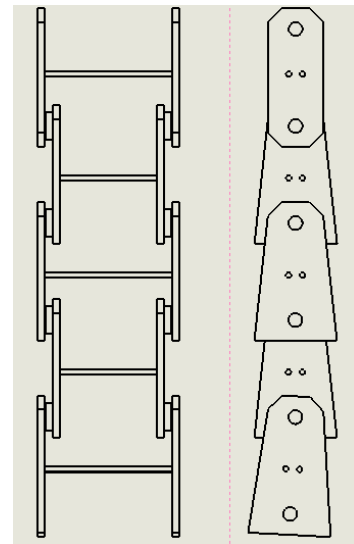


Imagen 4. Posición de las piezas para ensamble.

### 2.5. Análisis de movimiento

Para analizar el movimiento del robot debemos tener en cuenta los puntos de apoyo de los motores debido a que

en estos puntos se generan mayor presión al estar conectados directamente a las piezas, esto genera la onda sinusoidal de distribución de la fuerza de apoyo, la cual genera el mayor punto de esfuerzo a alcanzar los  $90^\circ$  y a iniciar el movimiento en  $0^\circ$ .

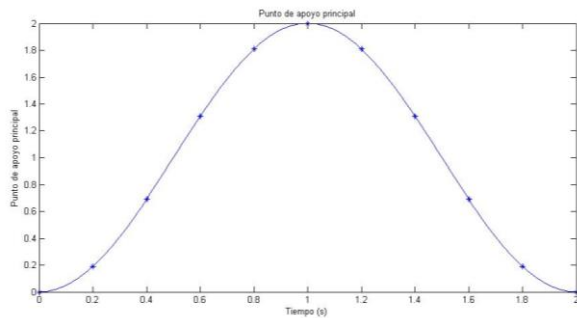


Imagen 5. Puntos de esfuerzo en el desplazamiento

Según la imagen 5 vemos el punto principal de apoyo de la celda unitaria del gusano donde se centra la mayor presión en la estructura a medida que se desplaza, al ser un movimiento uniforme encontramos que el desplazamiento puede ser descrito la mitad del periodo es

decir  $\frac{T}{2}$  que es equivalente a dos veces la longitud entre cada motor equivalente a  $2L$ , por lo tanto la

$$v = \frac{2L}{T/2} = \frac{4L}{T}$$

velocidad de desplazamiento es en la cual se puede deducir la ecuación de desplazamiento:

$$d = \frac{4L}{T} * t$$

Ecuación 1. Ecuación de desplazamiento

Con base a la ecuación encontrada anterior mente se realizó la siguiente grafica en la cual vemos como es la relación de tiempo y distancia sin tener en cuenta terrenos específicos que atrasaran el desplazamiento. En este encontramos que la relación de movimiento es estrechamente relacionada con los puntos de apoyo del motor y la distancia entre las aristas cosa que debe tener en cuenta.

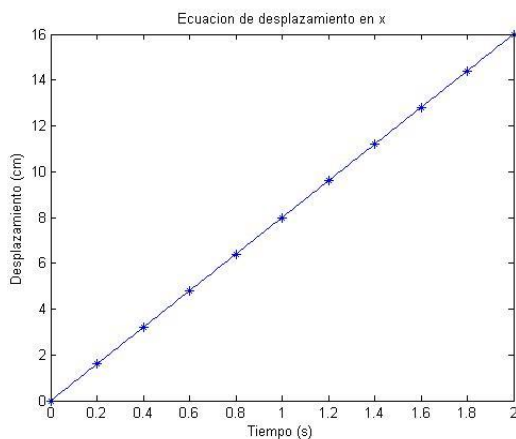


Imagen 6. Desplazamiento con base al tiempo

## 2.6. Análisis de esfuerzos

En la pieza anterior, podemos observar que ésta es sometida a fuerzas tanto externas como internas, las cuales producen un esfuerzo de la pieza a la hora de ejercer movimiento. Gracias a esto, la pieza sufre una deformación leve debido al diseño y fuerza ejercida por los motores al no tener amortiguadores.

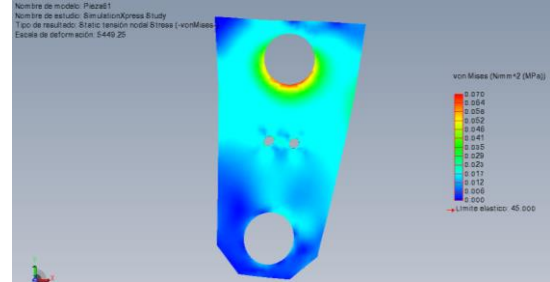


Imagen 7. Análisis de esfuerzos

En esta imagen vemos cómo la pieza sufre una deformación con base a un punto principal de apoyo, el cual sufre mayor esfuerzo (color rojo). Con base a este punto principal de apoyo se creó una deformación.

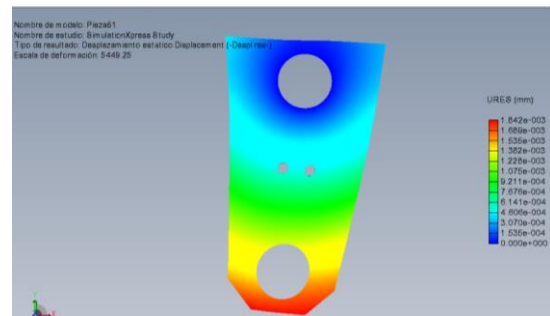


Imagen 8. Análisis de desplazamiento estático

En la imagen anterior observamos el nivel de desplazamiento de la figura con base al punto principal de apoyo, en este caso es el que está en color azul oscuro. Además, el punto de color rojo es un caso de mayor deformación y esfuerzo en un caso extremo, ya que este punto está unido a otra pieza unitaria, la cual funciona como un amortiguador mecánico, ocasionando que en otro caso en que se esfuerce mucho el robot, llegue a esta deformación.

## 3. CONTROL

Al analizar el funcionamiento mecánico nos dimos cuenta que se requiere precisión en los ángulos que se formaban entre los componentes para poder obtener el mejor rendimiento de robot, por lo que se escogió utilizar en la estructuras servomotores que aseguran esto gracias al pulso de PWM que se emplea en éste, también porque este motor proporciona una fuerza de torque considerable para mover una sección del robot y mantener esa posición.

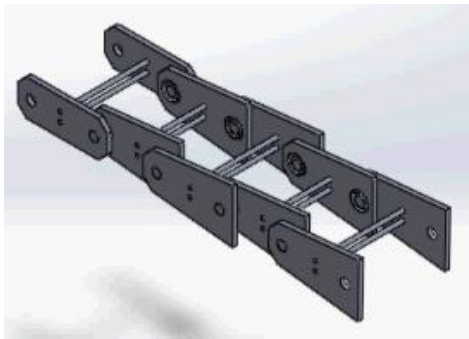
Para poder realizar el pulso de PWM para los servomotores se escogió una tarjeta arduino nano la cual

se enlazo a un módulo bluetooth y aun sensor de posición para así tener dos puntos de control para este.

Con base a esos dos puntos de control se realizó un programa el cual permite realizar el movimiento del gusano mientras no haya obstáculos a cierta distancia y el usuario dese que este se mueva y pueda monitorear la distancia de los obstáculos más cercanos.

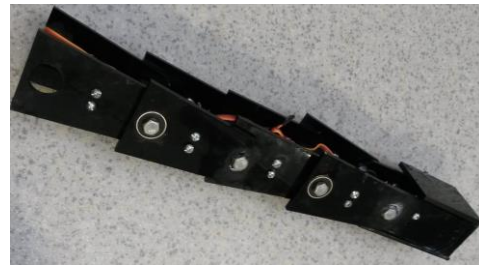
#### 4. RESULTADOS

Logramos analizar el movimiento rectilíneo del robot a través de la simulación mostrada en la imagen 9 la cual logramos ver que se planteó uniendo dos celdas unitarias las cuales ocasionaron que se doblara el desplazamiento mostrado pero manteniendo el cambio del punto de esfuerzos en cada celda la cual conforma la onda sinusoidal, con este cambio se tuvo en cuenta que al manejar varios motores aumenta la necesidad de mayor amperaje mientras que se maneja el mismo para el controlador el cual mantiene una fuente distinta de energía.



*Imagen 9. Simulación del gusano*

Después del análisis a través de la simulación seguimos a analizar el modelo físico de este mostrado en la imagen 10, al observar este encontramos que al no ser un movimiento gradual este generaba un golpe al cambiar de posición el cual se logra evaluar en el análisis de esfuerzos en el que se tomaron en cuenta las circunstancias en que se generaría mayor fuerza, en esta se generaría una deformación la cual no es muy significativa antes de regresar a su forma original, también logramos ver que en ciertas superficies no se generaba fricción entre los puntos de contacto entre el robot y el suelo por lo que se debió unir un tercer elemento entre estos unido al punto de contacto del robot para generar la fricción adecuada para el impulso de este.



*Imagen 10. Robot gusano*

#### 5. CONCLUSIONES

Debido a que esta clase de robot no ha sido explorado se analizó el movimiento de éste para poder ver sus cualidades, se limitó en un movimiento en dos dimensiones (2D) para facilitar su análisis y ver su comportamiento en el plano XY que son los principales ya que este mismo se puede replicar en el plano X,Z el cual complementa el movimiento y genera el desplazamiento en tres dimensiones (3D) por lo que éste no es necesario, encontramos que cambia el punto de masa como el punto de presión a medida que este se va desplazando formando así una onda sinusoidal a medida que cambia el ángulo del punto de apoyo por lo que se puede decir que mantienen una relación directa y que los puntos de mayor observación son los ángulos de  $0^\circ$  y  $90^\circ$  al ser en los que inicia el movimiento y el que se mantiene antes de terminar el ciclo para que inicia de nuevo.

Se debieron tener en cuenta circunstancias externas como los obstáculos, material del terreno y cantidad de objetos que se encuentran debido a que debido al desplazamiento del robot y a la estructura del programa este se detiene en la misma posición que tiene cuando está en reposo por lo que es necesario detener al robot a cierta distancia del obstáculo contemplando la distancia que este recorre suponiendo que se encuentra en su punto máximo que sería el escenario que propone mayor dificultad.

El robot al ser casi un autómata sigue su movimiento aun cuando este pierda la señal del usuario pero si se detiene cuando este lo requiere por lo que el papel del usuario es recibir los datos para conocer si hay un camino despejado y detenerlo si es requerido mientras se mantenga la señal, por eso hay que tener en cuenta los posibles factores que debiliten la señal a requerir una señal constante el usuario mas no es necesario para el robot.

#### 6. BIBLIOGRAFÍA

- [1]González Juan. 2008. Cube Revolutions, [http://www.learobotics.com/wiki/index.php?title=Cube\\_Revolutions](http://www.learobotics.com/wiki/index.php?title=Cube_Revolutions).
- [2](2012,Aug). Robot Gusano Meshworn. [http://www.ecured.cu/index.php/Robot\\_Gusano\\_Meshworn](http://www.ecured.cu/index.php/Robot_Gusano_Meshworn).
- [3] Granosik Grzegorz; Hansen Malik G; Borenstein Johann.2005. The OmniTread serpentine robot for

industrial inspection and surveillance.  
<http://search.proquest.com.ezproxy.unipiloto.edu.co/pqrl/docview/217013852/14021A274797CD3F9CB/12?accountid=50440#center>.

[4]Vergun, David.(2003, Jun). The slithering robot tries to worm its way into Navy's tactical units.  
<http://search.proquest.com.ezproxy.unipiloto.edu.co/pqrl/docview/235976991/14021A274797CD3F9CB/1?accountid=50440#center>.

[5]Anonymous.(1999, Apr). Biomorphie, robotic worm explorers tunnel. Ahead  
<http://search.proquest.com.ezproxy.unipiloto.edu.co/textgraphic/208826979/TextPlusGraphics/3?accountid=50440>.

## Anexo B

# ANÁLISIS CINEMÁTICO DE UN ROBOT ÁPODO

## Kinematic analysis of a snake robot

Jonathan Alexander Bermudez\*, Claudia Tatiana Martínez\*, Ruben Dario Hernández\*, Oscar Fernando Avilés\*,  
João Mauricio Rosario\*\*

\*Facultad de ingeniería. Programa de Ingeniería Mecatrónica. Grupo de Investigación Innovatic.

Universidad Piloto de Colombia. Carrera 9 # 45 A- 44 Bogotá, Colombia

Email: {jbermudez42, cmartinez99, ruben-herandez1, oscar-aviles}@upc.edu.co

\*\*Facultad de ingeniería mecánica, Laboratório de automação Integrada e Robótica

Universidade Estadual de Campinas -UNICAMP. Rua Mendeleiev, 200 cidade universitaria Campinas-SP, Brazil

Email: rosario@fem.unicamp.br

**Resumen**—Este artículo describe el análisis cinemático de un robot explorador del tipo ápedo, realizando su correspondiente modelado tanto en el espacio continuo como en el espacio discreto, con el objetivo de realizar la descripción del movimiento sinusoidal característico del robot. Además, se utiliza la cinemática directa y la cinemática inversa con el fin de comprender la funcionalidad y capacidades del robot ápedo.

**Index Terms**—Cinemática, cinemática directa, cinemática inversa, espacio continuo, espacio discreto, robot ápedo.

**Abstract**—This article describes the kinematic analysis of robot explorer type snake, performing a corresponding modelling both continuous space as in the discrete space with the objective to make the description of the sinusoidal movement characteristic of the robot. Additionally, the used the direct kinematics and the inverse kinematics in order to understand the functionality and capabilities of the snake robot.

**Keywords**—Kinematic, direct kinematics, inverse kinematics, continuous space, discrete space, snake robot.

### I. INTRODUCCIÓN

Los robots ápodos pertenecen a la rama de robots exploradores comprendida por la robótica móvil, esta clase de robot bio-inspirado es poco común a comparación de otros robots exploradores, gracias a que nace de la necesidad de explorar lugares estrechos como grietas y tuberías, en los cuales otros robots exploradores no podrían ser eficaces debido al tipo de locomoción que implementa o el tamaño requerido para lograr su desplazamiento por lo que no se podría utilizar en ese campo de acción. A diferencia de los otros robots exploradores, los robots ápodos al estar compuestos de celdas unitarias generan que su estructura requiera un espacio de trabajo mínimo gracias a las posibles distribuciones a lo largo de su estructura dando como resultado una buena adaptación en las áreas de exploración planteadas.

Las características del movimiento sinusoidal de los robots ápodos permiten movilizarse de forma eficaz en lugares reducidos. Una de las cualidades más destacadas del movimiento sinusoidal es su facilidad de análisis al encontrarse en dos dimensiones (2D) e interactuar entre planos gracias a la rotación de los módulos proporcionado un desplazamiento en tres dimensiones (3D) con base al mismo movimiento siendo limitado por las propiedades del material y diseño propio del robot.

Como expone Juan González Gómez[1] en su trabajo de «Principales líneas de investigación en robots tipo ápodos» o Alessandro Crespi, André Badertscher, André Guignard y Auke Jan Ijspeert [2] muestran algunos de los distintos tipos de robots ápodos diseñados para facilitar una tarea específica, como el robot «USAR» para rescate, el cual emplea una cámara posicionada en la parte frontal del robot para la transmisión de datos en lugares de difícil acceso o el robot «Snake» elaborado por Cornell Wright, Aaron Johnson, Aaron Peck, Zachary McCord, Allison Naaktgeboren, Philip Gianfortoni, Manuel Gonzalez-Rivero, Ross Hatton y Howie Choset [3] usado para la inspección en tuberías y alcantarillado. Esto se logra gracias a las cualidades de las celdas unitarias usadas para formar los robots, las celdas son iguales a lo largo de la estructura pero cambia según las cualidades deseadas para el desempeño adecuado en las tareas propuestas.

### II. MODELADO CINEMÁTICO DE UN ROBOT ÁPODO

El movimiento del robot ápedo es bio-inspirado al originarse en el movimiento sinusoidal característico de algunos animales que no poseen extremidades como son las serpientes y gusanos, los cuales se desplazan en lugares estrechos de difícil acceso como túneles y grietas gracias a la falta de necesidad de espacio que requiere para su desplazamiento junto con su capacidad de adaptabilidad al entorno.

El diseño de los robots ápodos para realizar el movimiento sinusoidal requiere una distribución de sus elementos a lo largo



de su estructura ocasionando la implementación de módulos para conformar el cuerpo del robot. Los módulos proporcionan un desplazamiento en dos dimensiones (2D) al estar fundamentados en un movimiento en dos dimensiones, pero al ser un robot del tipo explorador el movimiento requerido en tres dimensiones (3D) ocasionando entender la interacción entre los módulos ya que el giro de noventa grados (90°) entre los módulos permite el desplazamiento en el plano xyz como se expone en el trabajo de Juan González Gómez[4] o en el argumento expuesto en el trabajo de Andrés San Millán Rodríguez[5] titulado «DISEÑO, CONSTRUCCIÓN Y CONTROL DE UNA SERPIENTE ROBÓTICA» en la que también expone la transmisión de ondas sinusoidales con las cuales permiten la generación de curvas serpentinoides.

Para poder realizar el análisis cinemático del robot deben tener en cuenta los parámetros descritos continuación:

- **Ángulo de serpenteo ( $\alpha$ ):** Es el ángulo generado entre los segmentos del robot y un eje de referencia en el plano donde se refleja la onda sinusoidal.
- **Ángulo de desfase ( $\phi$ ):** Entiende el ángulo de desplazamiento entre la onda sinusoidal y el punto de referencia.
- **Número de módulos ( $M$ ):** Los módulos son la estructura principal que conforma el cuerpo del robot ápedo, el número necesario de módulos mínimo para la conformación de la estructura del robot es  $M \geq 2$  y el máximo será limitado según la tarea asignada por el usuario al variar el número de cables, batería, sensores entre otros.
- **Longitud del módulo ( $L$ ):** La longitud  $L$  está dada por la medida del módulo cuando está en reposo, es decir que el ángulo es 0°. Debido a que los segmentos son iguales, cada uno estará compuesto por 2 módulos ( $2L$ ).
- **Longitud del robot ( $l$ ):** Al ser construido por módulos iguales la longitud del robot se encontrará a través del producto entre la longitud de los módulos ( $L$ ) con el número de estos ( $M$ ) por lo que se obtiene la ecuación  $l = M \times L$ .
- **Distancia a lo largo de la curva ( $s$ ):** Es la distancia entre 0 y  $l$  a lo largo de la curva.
- **Número de ondulación ( $k$ ):** El número de ondulaciones es la cantidad de repeticiones de la onda sinusoidal con base a la longitud del robot.
- **Periodo ( $T$ ):** Es el tiempo tomado en realizar una ondulación. Tomando en cuenta los parámetros de longitud del robot ( $l$ ) y número de ondulaciones ( $k$ ) mencionados anteriormente se puede generar la ecuación  $T = \frac{l}{k}$ .
- **Número de articulación ( $i$ ):** Comprende el número de la articulación, del 1 hasta  $M$ .

Los anteriores parámetros pueden visualizarse en la *Figura 1*, en el cual expone el rango y la medida de los parámetros para su mejor comprensión.

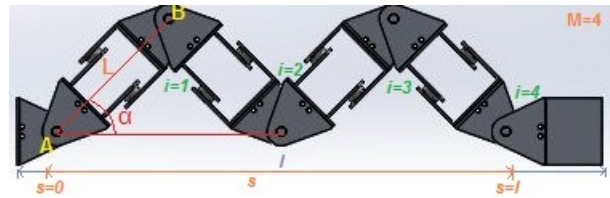


Figura 1: Visualización de parámetros. Fuente: Elaboración propia

La función sinusoidal esta dada por la ecuación 1.

$$F(x) = A \sin(b * x + c) \quad (1)$$

Remplazando con los parámetros considerando que es en base al espacio de posición «s» como describe Hadi Kalani, Alireza Akbarzadeh, Hosseinel Bahrami [6] da como resultado la ecuación 2.

$$K_{(s,\phi)} = -\frac{2\pi k}{l} \alpha \sin\left(\phi + \frac{2\pi k}{l} s\right) \quad (2)$$

Para encontrar las funciones que representen correctamente la onda de serpenteo es necesario encontrar la forma tangente a la curva de la ecuación 2.

$$\alpha_s = \alpha \cos\left(\phi + \frac{2\pi k}{l} s\right) \quad (3)$$

Con el fin de encontrar las coordenadas en el plano xy es necesario integrar «l» comprendida entre 0 y s de la ecuación 3 con base al coseno para la coordenadas en «x» y el seno en la coordenada en «y» encontrando las ecuaciones 4 y 5.

$$X_{(s,\phi)} = \int \cos\left(\alpha \cos\left(\phi + \frac{2\pi k}{l} * s\right)\right) ds \quad (4)$$

$$Y_{(s,\phi)} = \int \sin\left(\alpha \cos\left(\phi + \frac{2\pi k}{l} * s\right)\right) ds \quad (5)$$

Uno de los resultados de las ecuaciones encontradas anteriormente se pueden ver en la *Figura 2*, donde se observan tres ondas sinusoidales de color rojo, verde y azul que corresponden a noventa grados (90°), sesenta grados (60°) y cuarenta y cinco grados (45°) respectivamente.

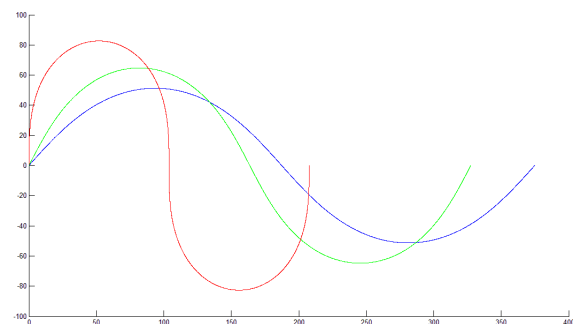


Figura 2: Variación de ángulo de serpenteo. Fuente: Elaboración propia

El ángulo mínimo de serpenteo está dado por la ecuación 6 obtenida de [1], debido al cambio del diseño del robot ápododo empleado como objeto de estudio y el ángulo de serpenteo máximo está dada por la ecuación 7 que no puede sobre pasar el ángulo de  $121^\circ$ . En dado caso que se sobrepasara el ángulo máximo ocasionaría un colapso entre los puntos de la curva. por lo que se debe tener en cuenta esta restricción.

$$\alpha_{\text{mínimo}} \approx \frac{120}{L} \left( \sqrt{\left(\frac{l}{L}\right)^2 + 18 - \frac{l}{L}} \right) \quad (6)$$

$$\alpha_{\text{máximo}} = \frac{A}{2 \sin\left(\frac{\pi * k}{M}\right)} \leq 121 \quad (7)$$

El ángulo máximo que pueden adoptar los robots ápodos es de ciento veintiuno grados ( $121^\circ$ ) mostrado en la *Figura 3* exponiendo una curvatura muy pronunciada, gracias a eso fue necesario emplear dos ondulaciones ( $k=2$ ) para mejor comprensión de su forma.

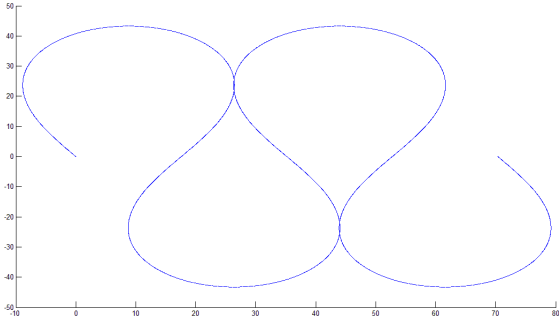


Figura 3: Onda de serpenteo con curvatura de  $121^\circ$ . Fuente: Elaboración propia

### III. MODELADO DISCRETO

El modelo discreto es la representación más acertada a la descripción de movimiento del robot ápododo, la variación del ángulo de serpenteo que es comprendida en función de la variable  $i$  que indica el número de articulación por lo que opta valores entre 1 hasta el número de módulos total ( $M$ ) como se muestra en la *Figura 4*.

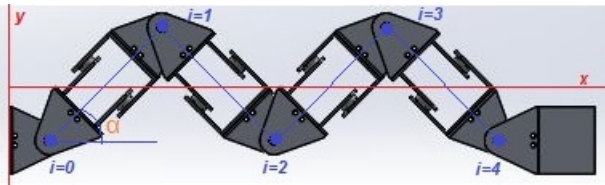


Figura 4: Gráfica en caso discreto. Fuente: Elaboración propia

Al cambiar a un plano discreto basado en el número del módulo  $i$  y el número de módulos ( $M$ ) es utilizado la ecuación continua 3 que da origen a la ecuación discreta 8.

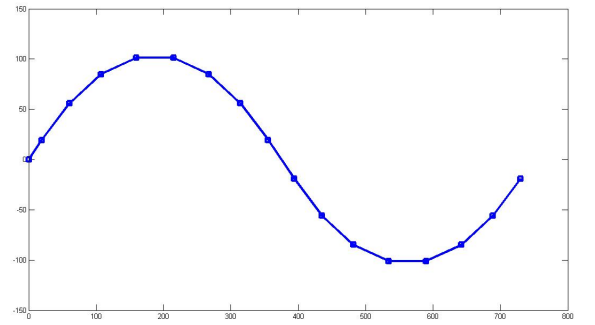
$$\alpha_i = \alpha \cos\left(\phi + \frac{2\pi k}{M}(i-1)\right) \quad (8)$$

Las coordenadas cartesianas  $xy$  en el plano discreto son comprendidas por las ecuaciones 9 y 10 que comprenden el cambio de posición de los módulos con base al anterior por lo que se realiza la sumatoria de la posición hasta  $M$  módulos, debido a que al iniciar el plano localiza el punto  $(0,0)$  y cambia la amplitud de la onda. Además, se requiere la implementación de ecuaciones compuestas.

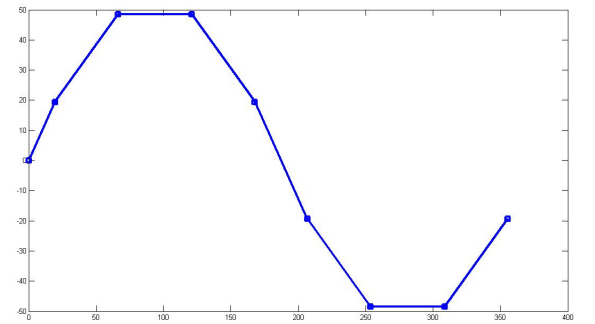
$$X_{(i)} = \begin{cases} \frac{L}{2} \cos \alpha & i = 1 \\ X_{(i-1)} + L \cos \alpha_i & i \in [2, M] \end{cases} \quad (9)$$

$$Y_{(i)} = \begin{cases} \frac{L}{2} \sin \alpha & i = 1 \\ Y_{(i-1)} + L \sin \alpha_i & i \in [2, M] \end{cases} \quad (10)$$

Utilizando las ecuaciones anteriormente expuestas (ecuaciones 9 y 10), se realiza la *Figura 5*, donde se realiza sólo una ondulación ( $k = 1$ ) y se cambió el número de módulos. Para la *Figura 5a* se implementaron 16 módulos ( $M = 16$ ) alcanzando así una significativa similitud con la onda de serpenteo, pero en la *Figura 5b* expone el posicionamiento de forma forzada ocasionando que no se pueda disminuir el número de módulos para alcanzar el posicionamiento para formar la onda sinusoidal completa.



(a) Posición 16 módulos



(b) Posición 8 módulos

Figura 5: Posición de los módulos. Fuente: Elaboración propia

### IV. CINEMÁTICA DIRECTA

La cinemática directa determina la posición del robot empleando una matriz de transformación homogénea como es expuesto por Jessica Hazdey Barreto Paz y Lizet Tatiana

Medina Ramírez[7] e Isaac Asimov[8], creando un sistema coordinado de ángulos de articulaciones causando que se requiera definir los parámetros de la matriz Denavit – Hartenberg, posteriormente se deben asignar los sistemas de referencia para así lograr calcular la transformada homogénea.

#### IV-A. Definición de los parámetros de la matriz Denavit – Hartenberg

Con base a la Figura 6 se definen los parámetros de la matriz de Denavit – Hartenberg. La locomoción del robot ápedo empleado para el análisis demuestra un movimiento limitado en el plano xy en el que exponen cinco articulaciones y cinco módulos.

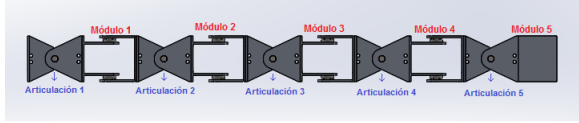


Figura 6: Modelo de robot ápedo. Fuente: Elaboración propia

En el cuadro I se definieron los parámetros donde L es la longitud del módulo y alpha es el ángulo formado entre los eslabones, no se considera la longitud en el eje z o rotación por el robot, debido a que el análisis es limitado a un sólo plano en dos dimensiones (2D), por lo que  $D_i = \theta_i = 0$ .

Articulación	$\theta_i$	$A_i$	$D_i$	$\alpha_i$
1	0	L/2	0	$\alpha_1$
2	0	L	0	$\alpha_2$
3	0	L	0	$\alpha_3$
4	0	L	0	$\alpha_4$
5	0	L	0	$\alpha_5$

Cuadro I: Parámetros Denavit-Hartenberg del robot ápedo. Fuente: Elaboración propia

#### IV-B. Asignación de los sistemas de referencia

La asignación del sistema de referencia se realiza a través de la transformación homogénea de cada articulación. Al ser de estructura simétrica entre los eslabones permite entender que las matrices resultantes puedan ser descritas por la matriz 12 a excepción de la matriz del primer eslabón expuesto en la matriz 11 al cambiar la longitud.

$$A_1 = \begin{bmatrix} \cos \alpha_1 & -\sin \alpha_1 & 0 & \frac{L}{2} * \cos \alpha_1 \\ \sin \alpha_1 & \cos \alpha_1 & 0 & \frac{L}{2} * \sin \alpha_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$$A_i = \begin{bmatrix} \cos \alpha_i & -\sin \alpha_i & 0 & L * \cos \alpha_i \\ \sin \alpha_i & \cos \alpha_i & 0 & L * \sin \alpha_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

#### IV-C. Cálculo de la transformada homogénea

Los cálculos de la transformada homogénea se realizan multiplicando la matrices homogéneas de los eslabones, es decir  $T = A_1 * A_i$  donde  $A_i$  comprende la multiplicación de las

matrices de los eslabones diferentes del primero, obteniendo como resultado la matriz 13 en la que los valores de la última columna describen la posición final.

$$T = \begin{bmatrix} \cos \alpha_i & -\sin \alpha_i & 0 & \frac{L}{2} \cos \alpha_1 + L \cos \alpha_i \\ \sin \alpha_i & \cos \alpha_i & 0 & \frac{L}{2} \sin \alpha_1 + L \sin \alpha_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

#### V. CINEMÁTICA INVERSA

Como expresa Samuel Muñoz González en su trabajo[9] y Javier Ollervides, Nadxielli Pineda, Armando Saenz, Víctor Santibáñez y Alejandro Dzul en [10], la cinemática inversa a diferencia de la cinemática directa emplea las coordenadas finales de posición en el plano para el posicionamiento del robot. La localización de los módulos se realiza a través de la relación geométrica de «La ley del coseno» y «El teorema de Pitágoras» mostrados en la Figura 7 en la que se encuentran dos módulos objeto de estudio con objetivo de simplificar las ecuaciones al repetirse a lo largo de la estructura.

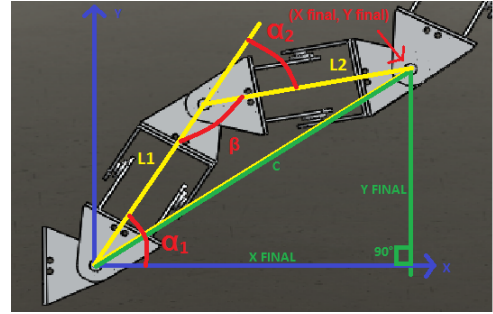


Figura 7: Posición entre módulos. Fuente: Elaboración propia

La ecuación 14 corresponde a «La ley del coseno» y la ecuación 15 a «El teorema de Pitágoras», empleando como base la Figura 7 se deduce que el valor denominado « $c^2$ » es igual en los dos casos por lo tanto se puede igualar las ecuaciones y reemplazando por las variables correspondientes al robot ápedo se obtiene la ecuación 16 en la que  $\beta = 180 - \alpha_2$  para obtener las variables objeto de estudio.

$$c^2 = a^2 + b^2 - 2ab \cos \beta \quad (14)$$

$$c^2 = X_{final}^2 + Y_{final}^2 \quad (15)$$

$$X_{final}^2 + Y_{final}^2 = (L_1)^2 + (L_2)^2 - 2L_1L_2 \cos(180 - \alpha_2) \quad (16)$$

Despejando el ángulo de  $\alpha_2$  se encuentra la ecuación 17.

$$\alpha_2 = \arccos \left( \frac{X_{final}^2 + Y_{final}^2 - (L_1)^2 - (L_2)^2}{2L_1L_2} \right) \quad (17)$$

Con el fin de encontrar la variable  $\alpha_1$  se emplea la ecuaciones 18 y 19 correspondientes a  $X_{final}$  y  $Y_{final}$  respectivamente.

$$X_{final} = L_1 \cos(\alpha_1) + L_2 \cos(\alpha_1 + \alpha_2) \quad (18)$$



$$Y_{final} = L_1 \sin(\alpha_1) + L_2 \sin(\alpha_1 + \alpha_2) \quad (19)$$

Al dividir los ángulos buscando mejor análisis de las ecuaciones se generan las ecuaciones 20 y 21.

$$X_{final} = \cos(\alpha_1)(L_1 + L_2 \cos(\alpha_2)) - \sin(\alpha_1)(L_2 \sin(\alpha_2)) \quad (20)$$

$$Y_{final} = \cos(\alpha_1)(L_2 \sin(\alpha_2)) + \sin(\alpha_1)(L_1 + L_2 \cos(\alpha_2)) \quad (21)$$

Con objetivo de despejar la variable  $\alpha_1$  se despeja de la ecuación 20 y reemplaza en la ecuación 21 obteniendo la ecuación 22.

$$Y_{final} = \frac{X_{final} + \sin(\alpha_1)(L_2 \sin(\alpha_2))}{(L_1 + L_2 \cos(\alpha_2))} (L_2 \sin(\alpha_2)) + \sin(\alpha_1)(L_1 + L_2 \cos(\alpha_2)) \quad (22)$$

Finalmente se deduce de la ecuación 22 respecto a la variable  $\alpha_1$  dando origen a la ecuación 23.

$$\alpha_1 = \arcsin \left( \frac{Y_{final}(L_1 + L_2 \cos(\alpha_2)) - X_{final}L_2 \sin(\alpha_2)}{X_{final}^2 + Y_{final}^2} \right) \quad (23)$$

## VI. RESULTADOS

Aplicando la cinemática directa se obtiene la *Figura 8*, en la cual se refleja los cambios del parámetro de alpha en la matriz de Denavit – Hartenberg entre 90° hasta el ángulo de 180° que le permite tomar su forma inicial de estudio en posición horizontal exponiendo un mejor control de posicionamiento.

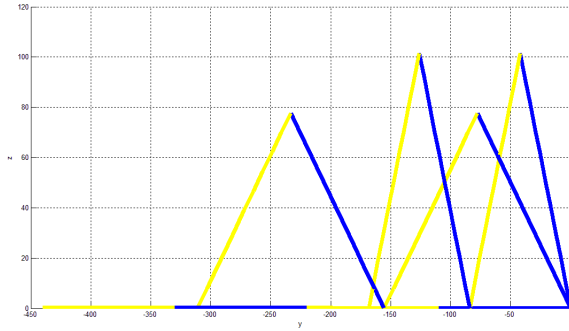


Figura 8: Resultado de cinemática directa comparación de movimiento. Fuente: Elaboración propia

Usando la cinemática inversa se crea la *Figura 9*, a pesar que se realizó el análisis para dos módulos sólo es requerido replicar los valores de los parámetros y así posicionar los módulos de forma simétrica, mostrando un control preciso con base al espacio a través de las coordenadas cartesianas.

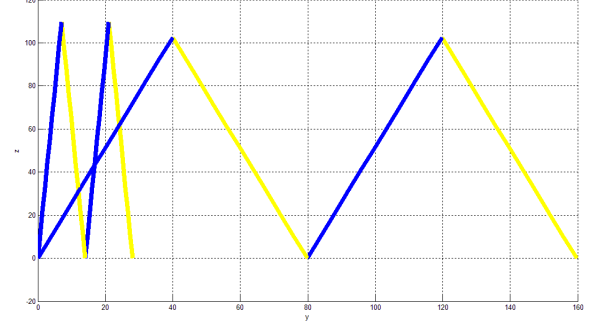


Figura 9: Resultado de cinemática inversa comparación de movimiento. Fuente: Elaboración propia

## VII. CONCLUSIÓN

Se puede decir que el análisis estático de la cinemática directa proporciona un mejor control del robot con base al cambio de posición con respecto al movimiento deseado ocasionando poder realizar movimientos robustos teniendo en cuenta las restricciones según el ángulo de serpiente y el rango permitido por las variables; pero al estudiar la cinemática inversa que emplea la coordenada en el espacio xy para su posicionamiento genera una necesidad de conocimiento preciso de las limitaciones y cualidades del robot debido a la inestabilidad frente a coordenadas inadecuadas para su posicionamiento.

## REFERENCIAS

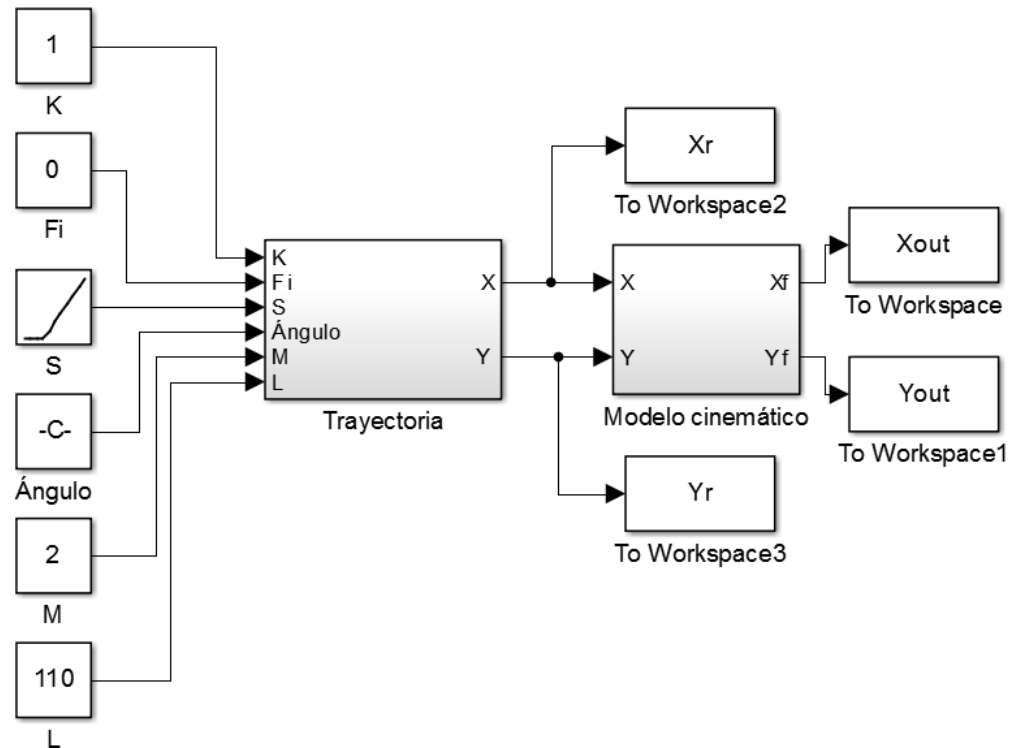
- [1] J. G. Gómez, "Principales líneas de investigación en robots tipo ápodos," 2002.
- [2] R. Crespi, A. Badertscher, A. Guignard, and A. J. Ijspeert, "2005a) swimming and crawling with an amphibious snake robot," 2005.
- [3] C. Wright, A. Johnson, A. Peck, Z. Mccord, A. Naaktgeboren, P. Gianfortoni, M. Gonzalez-rivero, R. Hatton, and H. Choset, "Design of a modular snake robot."
- [4] J. G. Gómez, "Robótica modular y locomoción: Aplicación a robots ápodos," *Universidad Autónoma de Madrid, Thesis*, 2008.
- [5] A. S. M. Rodríguez, "Diseño, construcción y control de una serpiente robótica," in *Creatividad, descubrimiento y futuro: I Congreso Nacional de Investigación en Grado INVEGRADO 2011, Albacete, Junio de 2011*. Universidad de Castilla-La Mancha, 2011, pp. 1385–1399.
- [6] H. Kalani, A. Akbarzadeh, and H. Bahrami, "Application of statistical techniques in modeling and optimization of a snake robot," *Robotica*, vol. 31, no. 4, pp. 623–641, 07 2013, copyright - Copyright © Cambridge University Press 2012; Última actualización - 2013-05-24. [Online]. Available: <http://ezproxy.unipiloto.edu.co/docview/1354767483?accountid=50440>
- [7] J. H. Barreto Paz and L. T. Medina Ramírez, "Simulador de trayectorias 2d para un robot tipo serpiente," 2010.
- [8] I. Asimov, "Locomoción en 2d."
- [9] S. Muñoz González *et al.*, "Análisis de la cinemática inversa y minimización de errores mediante algoritmos de optimización para un robot hiper-redundante," 2013.
- [10] J. Ollervides, N. Pineda, A. Saenz, V. Santibáñez, and A. Dzul, "Sistema de control de locomoción de un robot hexapodo caminante," 2013.

## Anexo C

### IDENTIFICACIÓN DE PLANTA DEL SISTEMA

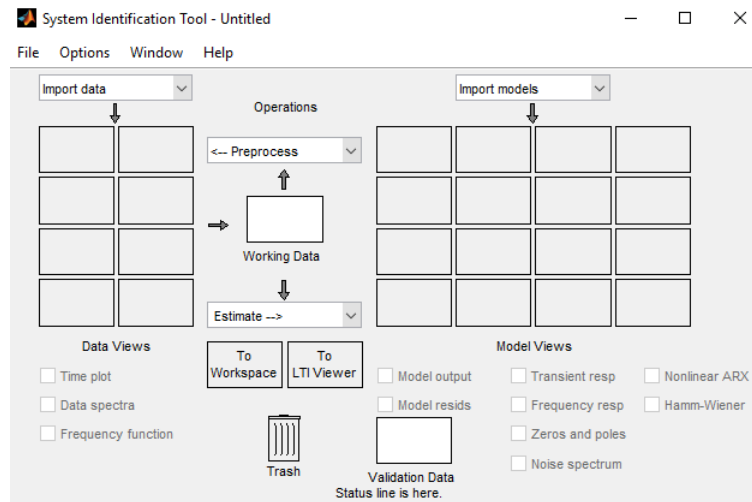
La identificación de la planta del sistema es realizada utilizando el «*toolbox*» de *Matlab®/Ident* considerando las entradas y salidas proporcionadas por el sistema, las cuales fueron encontradas a través de la (Figura 5.2) vista en el capítulo de «*RESULTADOS*» con ayuda del bloque «*To Workspace*» como se muestra en la (Figura 6.1) donde se exportan las variables  $Xr$  y  $Yr$  correspondientes a los parámetros de entrada junto a las variables de  $Xout$  y  $Yout$  que corresponden a la salida del sistema.

Figura 6.1: Diagrama de bloques usando «*To Workspace*»



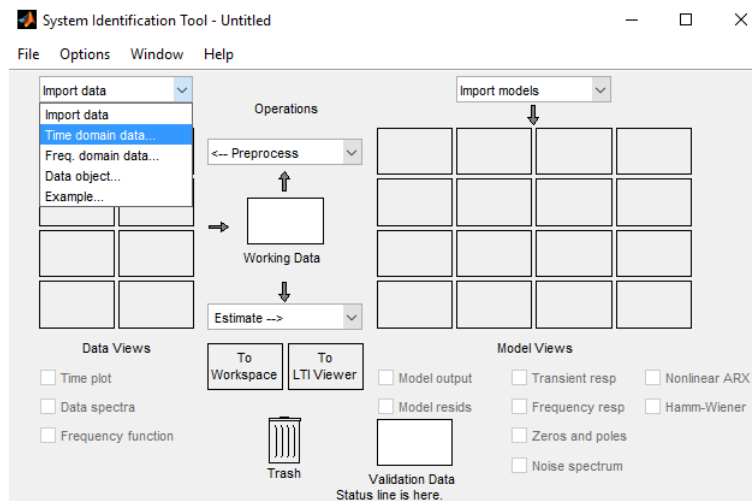
Fuente: Elaboración propia

Al utilizar el «*toolbox*» de *Matlab®/Ident* se encuentra la ventana presentada en la (Figura 6.2) donde se encuentra el cuadro de «*Import data*», «*Estimate*» e «*Import models*».

Figura 6.2: *Toolbox de Matlab®/Ident*

Fuente: Elaboración propia

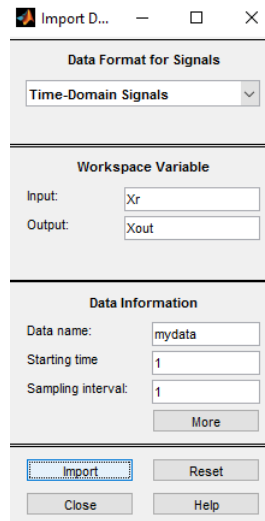
En el cuadro de «*Import data*» se selecciona la opción de dominio en el tiempo «*Time domain data...*» como se observa en la (Figura 6.3).

Figura 6.3: *Import data*

Fuente: Elaboración propia

Después se genera la ventana mostrada en la (Figura 6.4) donde son ingresados los datos de las variables de entrada y salida, en este caso son usadas las variables vistas anteriormente de  $X_r$  y  $X_{out}$  y posteriormente se debe presionar el botón «*Import*».

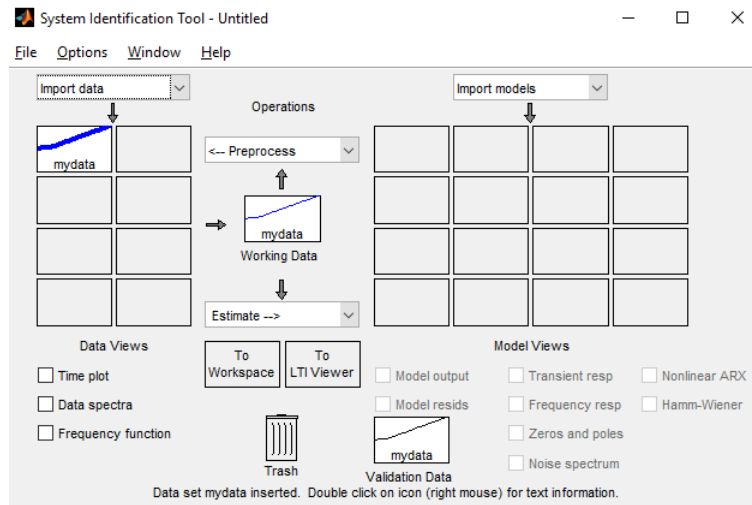
Figura 6.4: Datos de entrada



Fuente: Elaboración propia

De esta forma son importados los datos de entrada para realizar la identificación como se observa en la (Figura 6.5).

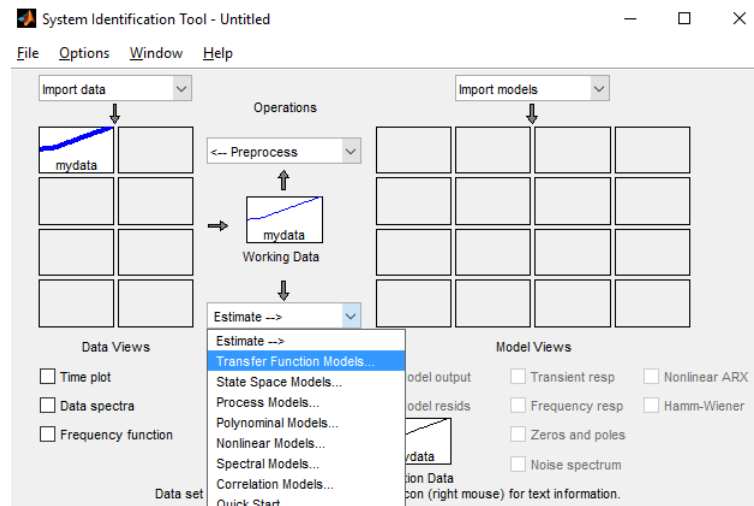
Figura 6.5: Identificación



Fuente: Elaboración propia

En el cuadro de «*Estimate*» es seccionada la opción de modelado para la función de transferencia «*Transfer Funtion Models...*» visto en la (Figura 6.6), enseguida aparece una ventada donde se indican el número de polos y ceros del sistema con el tipo de sistema deseado, en este caso son propuestos dos (2) polos y un (1) cero encontrados en el espacio continuo.

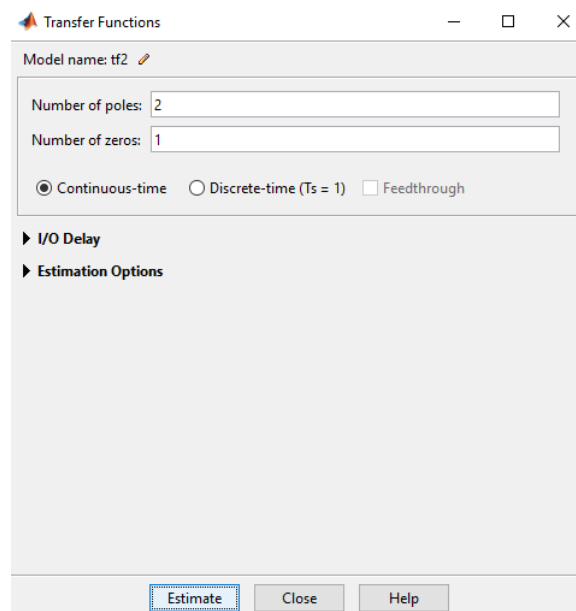
Figura 6.6: Modelado para la función de transferencia



Fuente: Elaboración propia

Después de definir los parámetros, se selecciona el botón de «*Estimate*» creando una ventana donde es identificado el progreso y algunos resultados como se observa en la (Figura 6.7).

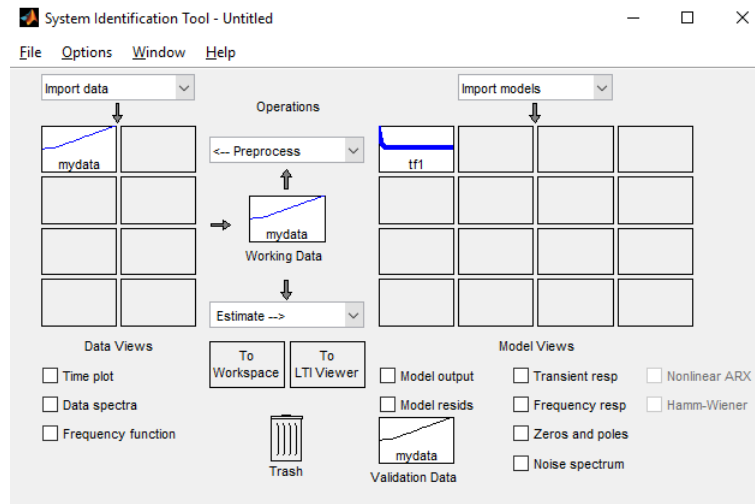
Figura 6.7: Proceso de estimación



Fuente: Elaboración propia

El resultado final se visualiza oprimiendo doble *click* en el recuadro resultante en el espacio «*Import models*» visto en la (Figura 6.8).

Figura 6.8: Resultado de modelo deseado



Fuente: Elaboración propia

Al realizar el procedimiento mostrado anteriormente se obtienen las funciones de transferencia encontradas en la ecuaciones 6.1 y 6.2.

$$Tfx = \frac{21,49 * S + 8,125}{S^2 + 11,39 * S + 8,118} \quad (6.1)$$

$$Tfy = \frac{-0,00239 * S - 0,0006237}{S^2 + 0,2265 * S + 8,982 * 10^{-8}} \quad (6.2)$$

## Anexo D

### DISEÑO Y APLICACIÓN DEL CONTROLADOR PID

Para realizar el diseño del controlador PID para el robot ápedo es necesario realizar la identificación del sistema en función de transferencia (visto en el «ANEXO C»). Realizando la identificación son encontradas las funciones expuestas en las ecuaciones 6.1 y 6.2 correspondientes a la función de transferencia para el eje X y eje Y respectivamente.

Utilizando el segundo método de sintonía de Ziegler-Nichols (mostrado en el capítulo dos) se realiza el diseño del controlador PID. Para realizar el diseño se usa el pseudocódigo mostrado en el algoritmo 6.1.

---

#### **Algoritmo 6.1** Ganancias y frecuencia para diseño del PID

---

**Programa:** Ganancias y frecuencia

**Entradas:** Función de transferencia

**Salida:** Gráfica localización polos y ceros

**Variables:** F

**Algoritmo**

F=función de transferencia

*toolbox* de gráfica de polos y ceros

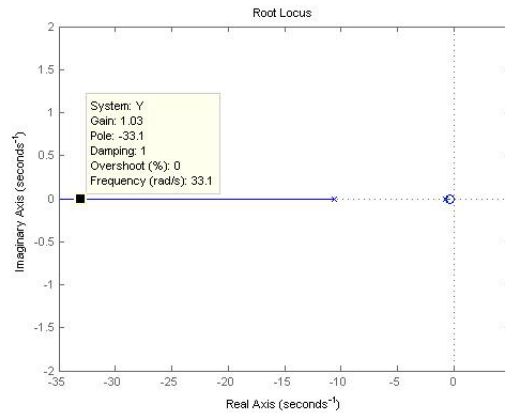
rlocus(F)

Fin del programa

---

En torno a la función de transferencia para la coordenada en el eje X mostrada en la ecuación 6.1 se utiliza el algoritmo 6.1 obteniendo la (*Figura 6.9*) donde se muestra el valor de la ganancia  $K_{cr} = 1,03$  y el valor de la frecuencia crítica  $P_{cr} = 33,17$  para obtener una respuesta sobre-amortiguada por lo que se obtiene los valores de las constantes de control  $Kd = 0,0147$ ,  $Ki = 6,5113$  y  $Kp = 0,618$  a través de las ecuaciones vistas en el cuadro 2.2 para controlador PID que usan el valor de las constantes de ganancia  $K_{cr}$  y frecuencia crítica  $P_{cr}$ .

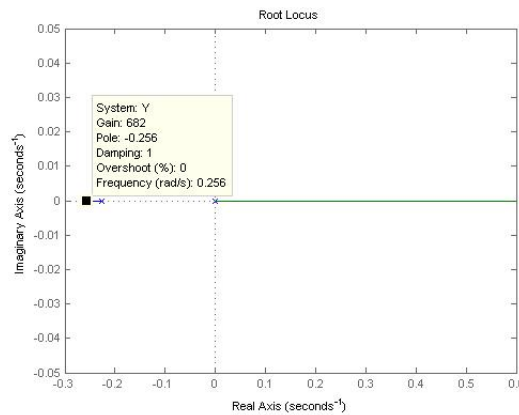
Figura 6.9: Ganancia y frecuencia de función de transferencia del eje X



Fuente: Elaboración propia

Para el controlador del eje Y se usa la función de transferencia para la coordenada en el eje Y vista en la ecuación 6.2 usando el algoritmo 6.1 encontrando la (Figura 6.10) exponiendo los valores de la ganancia  $K_{cr} = 682$  y el valor de la frecuencia  $P_{cr} = 0,256$  con los cuales se obtienen los valores de  $Kd = 1,2554 * 10^3$ ,  $Ki = 33,3446$  y  $Kp = 409,2000$  usando las ecuaciones ya mencionadas del cuadro 2.2.

Figura 6.10: Ganancia y frecuencia de función de transferencia del eje Y

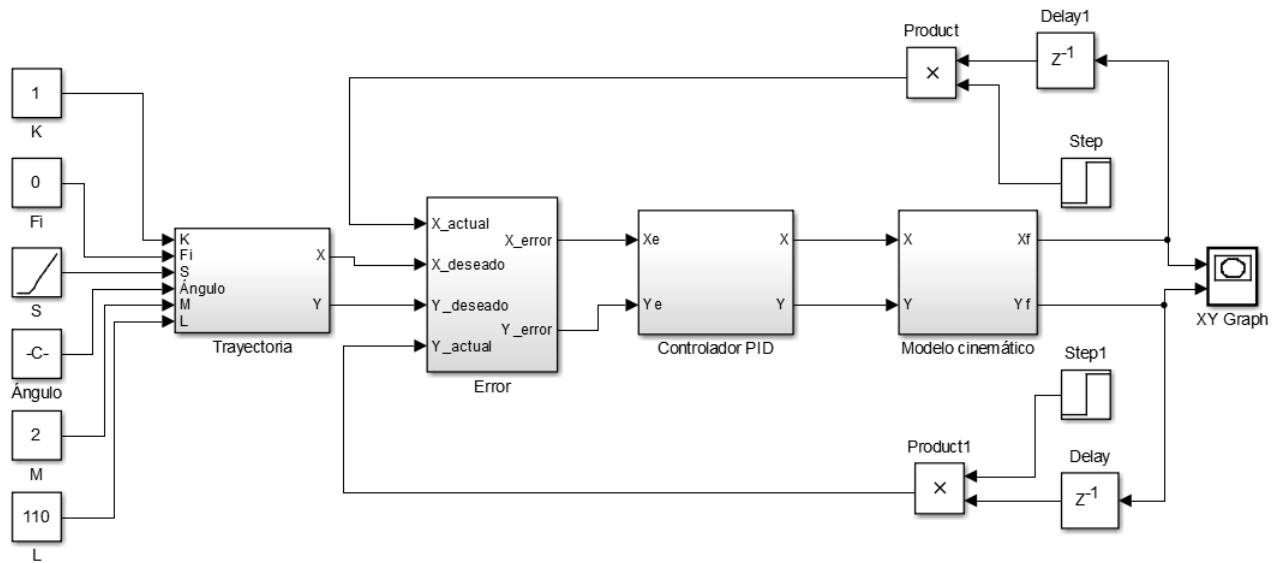


Fuente: Elaboración propia

En la (Figura 6.11) se observa el diagrama de bloques usado para la utilización del controlador PID que usa las constantes anteriormente encontradas.



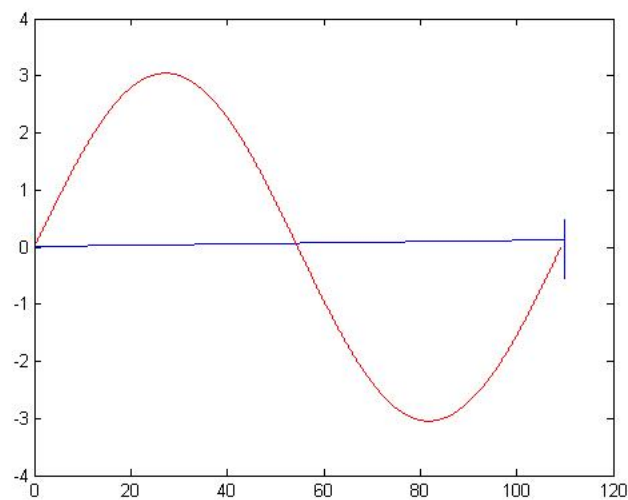
Figura 6.11: Sistema de control PID



Fuente: Elaboración propia

Usando el diagrama de bloques visto en la (Figura 6.11) utilizando las constantes de control PID se obtiene la respuesta vista en la (Figura 6.12) donde se refleja una respuesta no deseada de color azul junto con la trayectoria deseada de color rojo, por lo que se demuestra la ineficiencia del controlador frente al sistema utilizado debido a la cinemática empleada al ser diseñada para dos módulos, generando la necesidad de implementar un modelado matemático que contemple más módulos.

Figura 6.12: Respuesta del sistema con controlador PID



Fuente: Elaboración propia